Компьютерная лингвистика и интеллектуальные технологии

По материалам ежегодной международной конференции «Диалог» (2017)

Выпуск 16

Том 1 из 2
Компьютерная лингвистика:
практические приложения

Computational Linguistics and Intellectual Technologies

Papers from the Annual International Conference "Dialogue" (2017)

Issue 16

Volume 1 of 2
Computational Linguistics: Practical Applications

УДК 80/81; 004
ББК 81.1
К63

Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной международной конференции «Диалог» (Москва, 31 мая — 3 июня 2017 г.). Вып. 16 (23): В 2 т. Т. 1 — М.: Изд-во РГГУ, 2017.

Сборник включает 71 доклад международной конференции по компьютерной лингвистике и интеллектуальным технологиям «Диалог 2017», представляющих широкий спектр теоретических и прикладных исследований в области описания естественного языка, моделирования языковых процессов, создания практически применимых компьютерных лингвистических технологий.

Для специалистов в области теоретической и прикладной лингвистики и интеллектуальных технологий.

Предисловие

16-й выпуск ежегодника «Компьютерная лингвистика и интеллектуальные технологии» содержит избранные материалы 23-й международной конференции «Диалог». На основании мнений наших рецензентов для публикации в ежегоднике Редсоветом был отобран 71 доклад из числа примерно ста работ, которые были рекомендованы по результатам рецензирования для представления на конференции в 2017 году.

Работы в сборнике отражают все основные направления исследований в области компьютерного моделирования и анализа естественного языка, представленные на конференции:

- Компьютерные лингвистические ресурсы
- Компьютерный анализ документов (классификация, поиск, анализ тональности и т.д.)
- Корпусная лингвистика (создание, разметка, методики применения и оценка корпусов)
- Лингвистические онтологии и автоматическое извлечение знаний
- Лингвистический анализ Social media
- Лингвистический анализ речи
- Машинный перевод текста и речи
- Модели и методы семантического анализа текста
- Модели общения
- Теоретическая и компьютерная лексикография
- Типология и компьютерная лингвистика
- Формальные модели языка и их применение в компьютерной лингвистике

В соответствии с традициями «Диалога», старейшей и крупнейшей конференции по компьютерной лингвистике в России, отбор работ основывается на представлении о важности соединения новых методов и технологий анализа языковых данных с полноценным лингвистическим анализом и моделированием. Одной из важнейших целей конференции была и остается поддержка создания современных компьютерных ресурсов, моделей и технологий для русского языка.

В годовом цикле проведения конференции в рамках программы Dialogue Evaluation проводится тестирования технологий решения отдельных задач компьютерного анализа языка. На конференции подводятся итоги проведенных тестов, а статьи организаторов и наиболее успешных участников представляются в настоящем сборнике.

В этом году было проведено два тестирования:

1. По идентификации внешних заимстований (External Plagiarism Detection)
2. По оценке методов морфологического анализа русского языка, с акцентом на тексты Social Media.

Как обычно, результатом проведенных тестирований стали не только объективные данные о качестве работы различных методов и алгоритмов, но также и открытые для использования эталонные размеченные корпуса, т. н. золотые стандарты, позволяющие любым исследователям проводить сравнительные оценки эффективности своих технологий.

Все направления «Диалога» важны, но каждый год какие-то темы занимают особое место в программе конференции и в составе ежегодника. В этом году можно назвать две таких темы:

1. Применение методов глубинного машинного обучения: прежде всего — нейросетей и таких результатов их применения как word embeddings, как для прикладных задач, так и в лингвистических исследованиях.
2. В программе конференции этого года особенно заметны работы по использованию параллельных корпусов для лингвистических исследований. Такие корпуса уже давно и успешно используются в NLP, например, для обучения статистических моделей машинного перевода, автоматической дизамбигуации, автоматического построения языковых моделей. Но параллельные корпуса оказываются также и важным инструментом контрастивных лингвистических исследований.

Статьи в сборнике публикуются на русском и английском языках. При выборе языка публикации действует следующее правило:

- доклады по компьютерной лингвистике должны подаваться на английском языке. Это расширяет их аудиторию и позволяет привлекать к рецензированию международных экспертов.

- доклады, посвященные лингвистическому анализу русского языка, предполагающие знание этого языка у читателя, подаются на русском языке (с обязательной аннотацией на английском).

Несмотря на традиционную широту тематики представленных на конференции и отобранных в сборник докладов они не могут дать полной картины направлений «Диалога». Ее можно получить с помощью сайта конференции www.dialog-21.ru, на котором представлены обширные электронные архивы «Диалогов» последних лет и все результаты проведенных тестирований Dialogue Evaluation.

Мы обращаем внимание авторов и читателей сборника, что его бумажный вариант, который вы держите в руках, является вторичным по отношению к сборнику, который размещается на сайте конференции и индексируется Scopus. Мы рекомендуем при цитировании использовать именно сетевую версию.

Программный комитет конференции «Диалог»

Редколлегия сборника «Компьютерная лингвистика
и интеллектуальные технологии»

Организаторы

Ежегодная конференция «Диалог» проводится под патронажем Российского фонда фундаментальных исследований при организационной поддержке компании ABBYY.

Учредителями конференции являются:

- Институт лингвистики РГГУ
- Институт проблем информатики РАН
- Институт проблем передачи информации РАН
- Компания ABBYY
- Филологический факультет МГУ

Конференция проводится при поддержке Российской ассоциации искусственного интеллекта.

Международный программный комитет

Богуславский Игорь Михайлович	Институт проблем передачи информации РАН им. А. А. Харкевича, Россия
Буате Кристиан	Университет Джозефа Фурье — Гренобль 1, Франция
Гельбух Александр Феликсович	Национальный политехнический институт, Мехико
Иомдин Леонид Лейбович	Институт проблем передачи информации РАН им. А. А. Харкевича, Россия
Кобозева Ирина Михайловна	Московский государственный университет им. М. В. Ломоносова, Россия
Козеренко Елена Борисовна	Институт проблем информатики РАН, Россия
Корбетт Гревил	Университет Суррея, Великобритания
Кронгауз Максим Анисимович	НИУ «Высшая школа экономики», Россия
Лукашевич Наталья Валентиновна	НИВЦ МГУ им. М. В. Ломоносова, Россия
Маккарти Диана	Кембриджский университет, Великобритания
Мельчук Игорь Александрович	Монреальский университет, Канада
Нивре Йоаким	Уппсальский университет, Швеция
Ниренбург Сергей	Университет Мэриленда, Балтимор, США
Осипов Геннадий Семёнович	Институт системного анализа РАН, Россия
Раскин Виктор	Университет Пердью, США
Селегей Владимир Павлович	Компания ABBYY, Россия
Хови Эдуард	Университет Карнеги — Меллон, США
Шаров Сергей Александрович	Университет Лидса, Великобритания

Организационный комитет

Селегей Владимир Павлович, *председатель*	Компания ABBYY
Байтин Алексей Владимирович	Компания Yandex
Беликов Владимир Иванович	Институт русского языка им. В. В. Виноградова РАН
Браславский Павел Исаакович	Уральский федеральный университет
Добров Борис Викторович	НИВЦ МГУ им. М. В. Ломоносова
Захаров Леонид Михайлович	Московский государственный университет им. М. В. Ломоносова
Иомдин Леонид Лейбович	Институт проблем передачи информации РАН им. А. А. Харкевича
Кобозева Ирина Михайловна	Московский государственный университет им. М. В. Ломоносова
Козеренко Елена Борисовна	Институт проблем информатики РАН
Лауфер Наталия Исаевна	Компания Yandex
Ляшевская Ольга Николаевна	Институт русского языка им. В. В. Виноградова РАН
Толдова Светлана Юрьевна	НИУ «Высшая школа экономики»
Федорова Ольга Викторовна	Московский государственный университет им. М.В. Ломоносова
Шаров Сергей Александрович	Университет Лидса

Секретариат

Атясова Анастасия Леонидовна, *координатор оргкомитета*	Компания ABBYY
Белкина Александра Андреевна, *секретарь оргкомитета*	Компания ABBYY
Гусева Анна Александровна, *координатор Dialogue Evaluation*	Компания ABBYY
Севергина Екатерина Александровна, *администратор оргкомитета*	Компания ABBYY

Рецензенты

Августинова Таня
Антонова Александра Александровна
Азарова Ирина Владимировна
Андрианов Андрей Иванович
Апресян Валентина Юрьевна
Архангельский Тимофей Александрович
Байтин Алексей Владимирович
Баранов Анатолий Николаевич
Беликов Владимир Иванович
Бенко Владимир
Бердичевский Александр Сергеевич
Богданов Алексей Владимирович
Богданова-Бегларян Наталья Викторовна
Богуславский Игорь Михайлович
Бочаров Виктор Владиславович
Браславский Павел Исаакович
Васильев Виталий Геннадьевич
Галинская Ирина Евгеньевна
Галицкий Борис Александрович
Гельбух Александр Феликсович
Гецевич Юрий Станиславович
Гращенков Павел Валерьевич
Губин Максим Вадимович
Даниэль Михаил Александрович
Диконов Вячеслав Григорьевич
Добров Борис Викторович
Добровольский Дмитрий Олегович
Добрушина Нина Роландовна
Зализняк Анна Андреевна
Захаров Виктор Павлович
Захаров Леонид Михайлович
Ильвовский Дмитрий Алексеевич
Иомдин Борис Леонидович
Иомдин Леонид Лейбович
Катинская Анисья Юрьевна
Клышинский Эдуард Станиславович
Кибрик Андрей Александрович
Князев Сергей Владимирович
Кобозева Ирина Михайловна
Козеренко Елена Борисовна
Копотев Михаил Вячеславович
Коротаев Николай Алексеевич
Котельников Евгений Вячеславович
Котов Артемий Александрович
Кронгауз Максим Анисимович
Левонтина Ирина Борисовна
Лобанов Борис Мефодьевич
Лопухин Константин Александрович
Лукашевич Наталья Валентиновна
Лютикова Екатерина Анатольевна
Мисюрев Алексей Владимирович
Наков Преслав
Недолужко Анна Юрьевна
Падучева Елена Викторовна
Пазельская Анна Германовна
Паперно Денис Аронович
Панченко Александр Иванович
Переверзева Светлана Игоревна
Петрова Мария Андреевна
Пивоварова Лидия Михайловна
Пиперски Александр Чедович
Подлесская Вера Исааковна
Рахилина Екатерина Владимировна
Скулачева Татьяна Владимировна
Смирнов Иван Валентинович
Селегей Владимир Павлович
Слюсарь Наталия Анатольевна
Соколова Елена Григорьевна
Сомин Антон Александрович
Сорокин Алексей Андреевич
Сорокин Виктор Николаевич
Старостин Анатолий Сергеевич
Степанова Мария Евгеньевна
Тихомиров Илья Александрович
Толдова Светлана Юрьевна
Турдаков Денис Юрьевич
Урысон Елена Владимировна
Федорова Ольга Викторовна
Хохлова Мария Владимировна
Циммерлинг Антон Владимирович
Шаров Сергей Александрович
Шелманов Артём Олегович
Янко Татьяна Евгеньевна

Contents[1]

Приглашенные доклады

Компьютерная лингвистика: практические приложения

[*] Доклады упорядочены по фамилии первого автора в соответствии с английским алфавитом. The reports of each section are ordered by the surname of the first author in compliance with the English alphabet.

Приглашенные доклады

OPEN KNOWLEDGE REPRESENTATION FOR TEXTUAL INFORMATION

Ido Dagan (ido.k.dagan@gmail.com)

Bar-Ilan University, Israel

How can we capture the information expressed in multiple texts? And how can we allow people, as well as computer applications, to easily explore it? When comparing textual knowledge to formal knowledge representation (KR) paradigms, two prominent differences arise. First, typical KR paradigms rely on pre-specified vocabularies, which are limited in their scope, while natural language is inherently open. Second, in a formal knowledge base each fact is encoded in a single canonical manner, while in multiple texts facts may be repeated with some redundant, complementary and even contradictory information.

In this talk, I will outline a new research direction, which we term Open Knowledge Representation (OKR), that aims to represent textual information in a consolidated structured manner, based on the available natural language vocabulary and structure. I will describe our first specification for Open Knowledge Graphs, motivated by a use case of representing multiple tweets describing an event, for which we have created a medium-scale annotated dataset. Our proposed structure merges co-referring individual proposition extractions, created in an Open-IE flavor, into a representation of consolidated entities, predicates and statements, inspired by traditional knowledge graphs. Information redundancy is further modeled via entailment relations. I will also illustrate the potential application of our open knowledge graphs for text exploration and point at possible directions in which the OKR paradigm might evolve.

DEEP LEARNING AND LANGUAGE ADAPTATION

Sergey Sharoff (S.Sharoff@leeds.ac.uk)

University of Leeds, UK

Many lesser-resourced languages are related to languages, which have better resources. For example, the Universal Dependency treebank has about 2 MW of training resources for Czech, more than 1 MW for Russian, while only 950 words for Ukrainian and nothing for Belorussian, Bosnian or Macedonian. Similarly, the Autodesk Machine Translation dataset only covers three Slavonic languages (Czech, Polish and Russian). In this talk I will discuss a general approach, which can be called Language Adaptation, similarly to Domain Adaptation. In this approach language models can be adapted from a better-resourced (donor) language to a lesser-resourced (recipient) language. In my talk I will discuss examples of a Deep Learning architecture for Language Adaptation, which is based on creating a shared representation across related languages. Three case studies will be presented: Part-Of-Speech tagging, Named Entity Recognition and Translation Quality Estimation. I will also discuss the importance of the typological distance between the donor and the recipient.

Компьютерная лингвистика: практические приложения

PART-OF-SPEECH TAGGING WITH RICH LANGUAGE DESCRIPTION

Anastasyev D. G. (daniil_an@abbyy.com),
Andrianov A. I. (andrew_an@abbyy.com),
Indenbom E. M. (eugene_i@abbyy.com)

ABBYY, Moscow, Russia

This paper deals with morphological parsing of natural language texts. We propose a method that combines comprehensive morphological description provided by ABBYY Compreno system and sophisticated machine learning techniques used by the state-of-the-art POS taggers. The morphological description contains information about possible grammatical values of a dictionary word that helps to identify a set of potential hypothesis for each word during the morphological analysis stage. To analyse out-of-vocabulary words we are building a number of most likely paradigms in the morphological model using the orthographic features of the analysed word. The proposed method helps to reduce the number of hypotheses using the context information of each word. We use Bidirectional LSTM classifier to handle the context information and to predict the most probable grammatical value. The ambiguous grammatical values obtained from morphological description are used as features for the classifier. Also, we use word embeddings and orthographic features to achieve better results.

Key words: pos-tagging, morphological analysis, lemmatization, machine learning, lstm

МОРФОЛОГИЧЕСКАЯ РАЗМЕТКА С ИСПОЛЬЗОВАНИЕМ ОБШИРНОГО ОПИСАНИЯ ЯЗЫКА

Анастасьев Д. Г. (daniil_an@abbyy.com),
Андрианов А. И. (andrew_an@abbyy.com),
Инденбом Е. М. (eugene_i@abbyy.com)

ABBYY, Москва, Россия

1. Introduction

Part-of-speech (POS) tagging is the task of assigning each word in the given text an appropriate grammatical value. The morphological analysis is an essential element of most NLP problems. It means that quality of their solutions highly depends on the quality of the POS tagging.

Most researches on POS tagging have focused on English. The evaluation of these models is typically based on Penn Treebank. The latest approaches have claimed to achieve more than 97.55% accuracy. Unlike the previous methods, the newest ones are usually designed to use as few morphological features as possible. Such solutions are more likely to have stable performance on different corpora and to have the ability to be trained on various languages.

The most well-known Russian POS taggers are mystem [Segalovich, 2003], TnT-Russian [Sharoff, Nivre, 2011], Tree-Tagger [Schmid, 1994]. In contrast to the modern English taggers, the mentioned algorithms mostly rely on morphological features. However, their comparison is difficult because of lack of standard morphological tagset and corpora for the Russian language for tagger evaluation.

This work aims to combine comprehensive morphological description provided by ABBYY Compreno system [Anisimovich et al., 2012] and quite sophisticated machine learning techniques used by the novel English POS taggers.

Its evaluation was performed during the RuMorphoEval-2017 competition which was designed to provide a standard tagset and corpus for taggers comparison purposes.

2. Proposed Method

2.1. Russian Morphological Model

The most notable distinction of the proposed approach is the usage of the rich morphological model of the Russian language. It consists of a vast number of morphological paradigms and extensive lexicon. The dictionary consists of about 240 thousand of lexemes which provide us more than 3.5 million of words.

Such a significant number of words could be stored quite compactly based on the information about the words' paradigms. These paradigms contain information about the grammatical value of dictionary word and its inflexion. Therefore, we can store in the dictionary only the lexemes and the paradigms and obtain the needed word by composing this information. Overall, there were identified more than three thousand Russian paradigms.

As a result, usually, the words in corpora can be found in the lexicon and analysed by the provided morphological model. However, this analysis is not unambiguous: most of the words are homonymous.

This ambiguity may take place between the words of the same lexeme. For instance, “стол” (“table”) can be either nominative or accusative form. Also the ambiguity can appear between the words of different lexemes: “стекло” may be both noun (“glass”) and verb (“to flow down”).

To deal with the ambiguity, we need to use context information. In the next sections, we are going to describe the method used to choose the correct analysis.

2.2. Unknown Words Processing

Despite the size of the provided lexicon, there are many out of vocabulary words in the texts. The most obvious examples of such words are named entities and neologisms.

To lemmatize such words, we use the following technique. We are constructing a set of pseudo-forms—hypothetical analyses of the given word. Then we are sorting them by their quality—the probability that the word is in such paradigm.

During the first step, we have to obtain all pairs of stem and paradigm conformed to our language model. As a result, we are going to receive the grammatical value of the word and its inflexion.

The stem of the word is its part without ending. All potential endings of the word can be found in the language description. Moreover, for each ending we can collect all possible paradigms—that is all paradigms where such ending occurs. Therefore, we get few stems with a limited number of paradigms agreed with the stem according to the language model.

Thus, we build a set of hypotheses—more than half of thousand in average. The next step is their ranking. The key element of the sorting is the usage of N-gram statistics of suffixes of word's stems. It based on the assumption that new words should contain patterns similar to some fragments of existing ones. Then it is likely to find these patterns in the suffixes of dictionary words.

Therefore, we should prefer forms that maximise the following function:

$$Q(form) = \mathrm{P}(paradigm(form), suffix(form))$$

Such probability can be estimated by corpora information.

Still, to improve the ranking, we should use the context information in a similar way as in the case of dictionary words.

2.3. Features

In our model following features are incorporated.

Grammatical value. Obviously, the information about the grammatical values of context words is vital in determining the grammatical value of the analysed word. We store these grammatical values of a word in the vector of size equals to the overall number of available grammemes. It means that each component of this vector corresponds to some grammeme.

However, as was mentioned in section 2.1, practically each morphological analysis contains some homonymy. So we write into the vector the estimated probability of each grammeme. The probability is calculated using the sum of frequencies of the morphological forms contains such grammeme.

For instance, consider the word “стул” (“chair”). It is a nominative form with frequency equals to $1.03 \cdot 10^{-6}$ or accusative form $8.15 \cdot 10^{-7}$ frequency. Thi leads us to the quality of the accusative grammeme calculated as

$$\frac{8.15 \cdot 10^{-7}}{8.15 \cdot 10^{-7} + 1.03 \cdot 10^{-6}} \approx 0.4417$$

Ambiguity classes' probabilities. Another type of features is probabilities of predicted classes (i.e. word's possible grammatical values). Such features set soft constraints on predictions. The probability is proportional to max frequency of form obtained by morphological analysis of the word.

Punctuation. The binary feature that corresponds to whether particular punctuation mark appears in the particular position in the word's surrounding.

Word's case type. The binary feature that says whether the word has proper, or upper, or lower capitalization.

Suffixes. The binary feature: whether the word has such suffix. Suffixes with length up to 3 were used during the developing of the model. To reduce the dimensions of the feature space, suffixes with low frequency were pruned: we collected 35 one-letter suffixes, 507 two-letters suffixes and 2316 three-letters suffixes with considerably large frequency.

Word Embeddings. 250-dimension dense vector corresponding to some word. The word embeddings technique has proved to be very effective in various NLP-tasks. There is a number of state-of-the-art English POS-taggers which utilise the power of the technique.

2.4. Learning Model

Predicted Classes. We enumerated grammatical values encountered in the train set. It appears to be slightly less than three hundred different categories of grammatical values. Hence, we can formulate the aim of the learning algorithm as a multiclass classification between the obtained grammatical values.

In this paper, to use a context of the analysed word, we take advantage of the Bidirectional Long-Short Term Memory neural networks (BiLSTM) [Hochreiter and Schmidhuber, 1997].

LSTM Classifier. LSTM is a variant of recurrent neural networks (RNNs). The RNNs use the information from the previous predictions to choose the label of the current input. Such architecture suits to POS-tagging better than traditional neural networks. However, it was proved that RNNs suffer from the gradient vanishing problem [Bengio et al., 1994]. It means that the ordinary recurrent network is aware only about the inputs from the short-period, but the information from more time steps is vanishing. LSTMs use gating mechanism to deal with the problem. It helps to the network to explicitly model long-term dependencies.

Meanwhile, the LSTM's hidden state stores information only about the previous words. To obtain the data from both the previous and the next words, we use the Bidirectional LSTM architecture. Its basic idea is to combine two LSTMs—forward and backward—and concatenate their output. Such a simple solution has been proven to be effective in the POS-tagging and similar tasks.

In this work, we decided to use a two-layer Bidirectional LSTM. During the development of the model, the additional layer gave obvious improvements in the performance of the LSTM on the validation set. However, it should be noted that such improvement may not be necessary for the practical usage. With the extra layer, both the train and the prediction time increases twofold. Moreover, the size of the network grows up. As a result, the usage of the second layer does not seem to be mandatory.

Additional Layers. Furthermore, we add a hidden Dense layer with ReLU activation on top of the LSTMs. This layer should help to handle the nonlinearity of the problem. The ReLU activation function is designed to deal with gradient vanishing problem. To connect this layer with the LSTM, we use the TimeDistributed wrapper from the keras library. This wrapper is used to apply the Dense layer to each word in the sentence separately.

Output Layer. The output layer is also wrapped by the TimeDistributed layer and it uses softmax activation function to output the probabilities for each considered grammar value.

Input Layers. We use few distinct input layers. First of all, we have a Grammemes input layer. It receives morphological features—the grammatical value, ambiguity classes' probabilities and the word's case type—and information about punctuation. Overall, we collected 617 different features. It is much lesser than the number of features used in most of the state-of-the-art classifiers. The main reason to such small set is the specificity of neural networks: we hardly can train a network on a large and sparse feature set. On the other hand, the features obtained by morphological analysis seem to be strong enough to rely on them.

As stated in section 2.3, we also utilise word embeddings technique. So we have another input layer to perform it. The model with both word embeddings and morphological features dramatically outperformed the model that uses only the morphological features.

We have considered the usage of the suffix features. We apply them using the embedding technique: for each suffix length we create a separate input layer and pass the input to the embeddings layer.

To reduce the dimensions of word embeddings' and suffixes' features, we implement a preprocessing to each analysed word. We substitute by a star ('*') all letters that do not belong to Russian alphabet or number, punctuation and symbols Unicode character categories. We replace each digit by zero ('0'). Finally, we convert each word to lower case. Such normalization leads to the reduction of the number of possible different word and suffix types.

Regularization. For the regularisation proposes we use Dropout technique [Srivastava et al., 2014]. We apply dropout to the Embedding layer, to the output of the LSTMs and inside the LSTM layers. Also, we utilise Batch Normalization [Ioffe, 2015] for the hidden Dense layer. This method helps to achieve faster learning speed and higher overall accuracy.

Optimizer. As an optimisation algorithm we have chosen the Adam optimizer [Kingma and Ba, 2014].

Summary. We implemented our model using the keras library[1] on theano backend [Bergstra et al., 2010].

The Fig. 1 illustrates the basic structure of our neural network with parameters corresponded to the keras parameters.

[1] From keras library: https://github.com/fchollet/keras/

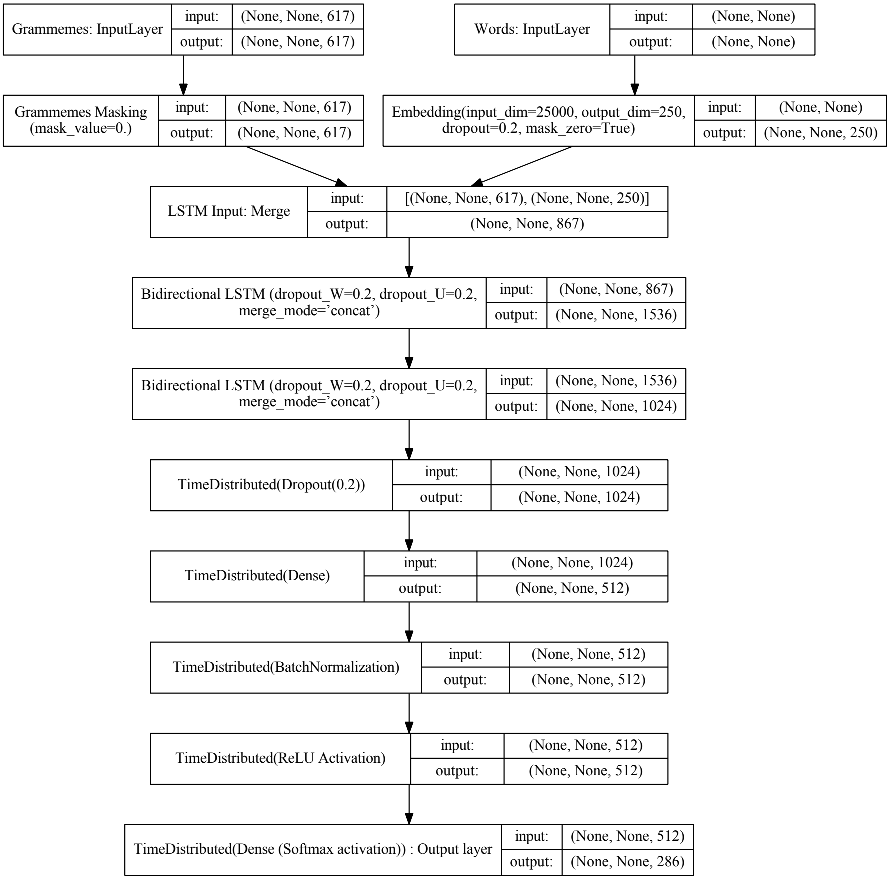

Grammemes: InputLayer
input: (None, None, 617)
output: (None, None, 617)
Words: InputLayer
input: (None, None)
output: (None, None)
Grammemes Masking (mask_value=0.)
input: (None, None, 617)
output: (None, None, 617)
Embedding(input_dim=25000, output_dim=250, dropout=0.2, mask_zero=True)
input: (None, None)
output: (None, None, 250)
LSTM Input: Merge
input: [(None, None, 617), (None, None, 250)]
output: (None, None, 867)
Bidirectional LSTM (dropout_W=0.2, dropout_U=0.2, merge_mode='concat')
input: (None, None, 867)
output: (None, None, 1536)
Bidirectional LSTM (dropout_W=0.2, dropout_U=0.2, merge_mode='concat')
input: (None, None, 1536)
output: (None, None, 1024)
TimeDistributed(Dropout(0.2))
input: (None, None, 1024)
output: (None, None, 1024)
TimeDistributed(Dense)
input: (None, None, 1024)
output: (None, None, 512)
TimeDistributed(BatchNormalization)
input: (None, None, 512)
output: (None, None, 512)
TimeDistributed(ReLU Activation)
input: (None, None, 512)
output: (None, None, 512)
TimeDistributed(Dense (Softmax activation)) : Output layer
input: (None, None, 512)
output: (None, None, 286)

Fig. 1. Structure of the neural network

3. Model Development

The model was trained during participation in the MorphoRuEval-2017 competition[2]. In this competition the multiclass accuracy is used as the metric.

3.1. Tagset

The competition used slightly modified Universal Dependencies tagset[3].

Our morphological description is based on other tags, so we wrote a converter from our grammatical values to the required tagset. The convertor's mapping sets the

[2] https://github.com/dialogue-evaluation/morphoRuEval-2017

[3] http://universaldependencies.org/ru/feat/all.html

one-to-many relationship between our grammemes and the grammemes in the Universal Dependencies. It means that in some cases we convert our grammatical value to a few Universal Dependencies grammatical values with frequencies equal to the frequency of the initial grammatical value.

The converter is used in two ways. First of all, it is applied to obtain the set of ambiguity classes' probabilities. It seems acceptable to have an additional ambiguity due to the conversion process. Besides, we used the converter to train our model on additional corpora that were tagged with our tagset.

3.2. Training Data

As an additional corpora, we used a subset of Russian Wikipedia and parallel corpus of translated English novels. The Wikipedia corpus contains more than 3 million tokens. From the corpus of novels, we extracted subcorpus with about 30 million tokens. We used ABBYY Compreno system to perform tagging of the texts.

Besides, we used GICR texts with Universal Dependencies tagset[4]. This corpus consists of about one million tokens. It contains sentences from different social media sources.

3.3. Sentences padding

We use LSTMs to be able to deal with the whole sentence during classification stage. However, this neural network requires a three-dimensional tensor as an input. Due to inequality of the sentence lengths, we are not able to store the train data in one tensor without any changes. One of such possible changes is padding method: we choose the maximum sentence length and pad (i.e. add zeros to) all shorter sentences.

In addition to padding, we use a masking mechanism: we restrict the network from training on the padded elements.

The padding may drastically expand the size of the train data: with large maximum sentence length, we would usually waste memory on the zeros in the short sentences. To reduce the usage of memory, we divided all sentences into a few groups with different length: the sentences with up to 6, from 7 to 14, from 15 to 25, from 26 to 40 and more than 40 words.

3.4. Word Embeddings

As a baseline, we used randomly uniformly initialized embeddings for the first 5000 most frequent words with output dimension equals to 250. Surprisingly, the pretrained word embeddings (about 470 thousand words and 200-dimension output vector) had not given any enhancement. We decided to use randomly initialized embeddings of the 25 thousand most frequent words only.

[4] https://github.com/dialogue-evaluation/morphoRuEval-2017/blob/master/GICRYA_texts.zip

3.5. Model Training

To evaluate the quality of the model, we divided the GICR data into train and validation set at a ratio of 2 to 1.

The model development was performed in the following way. Firstly, we have experimented on the GICR texts to obtain optimal network architecture and the best set of hyperparameters. Then we have used the additional corpora to improve the achieved results. Also, we have experimented with extra layers and increased number of neurones on the additional data.

During the first stage, we found out the effectiveness of the network's architecture described in Fig 1. We achieved 96.31% accuracy on the validation set.

The additional suffixes features increased the accuracy up to 96.41%.

The usage of the extra Wikipedia subcorpus helped to improve the classifier quality to 96.78%. At the same time, the model trained on the Wikipedia only managed to achieve only 93.24%. The reason for such poor quality seems to be the case of the known fact: the accuracy of tagger trained on one text genre drops dramatically on other genres [Giesbrecht and Evert, 2009].

To deal with the problem, we applied the technique known as fine tuning. We used the weights of the model pretrained on the Wikipedia to initialize weights of the model and trained the model on GICR. That led to 97.43% accuracy.

We exploited this method to train the model on our novels subcorpus. The model trained on the novels subcorpus only was able to reach 95.36% accuracy. Fine tuning of this model on the GICR texts gave 97.78% accuracy, which is our best result on the validation set.

The Table 2 summarises the performance of the model achieved by usage of different train sets.

Table 2. Accuracies of the model trained on different corpora achieved on the validation set

Model	Train corpus	Accuracy on the validation set
Basic model	GICR	96.31%
+ suffix features	GICR	96.41%
+ suffix features	Wiki	93.24%
+ suffix features	GICR + Wiki	96.78%
+ suffix features	pretrained on Wiki, trained on GICR	97.43%
+ suffix features	Novels	95.36%
+ suffix features	pretrained on Novels, trained on GICR	**97.78%**

4. Evaluation

The evaluation was performed on three different genres of texts: fiction texts[5], news texts[6] and social networks texts[7].

4.1. Achieved Results

Our system received the following results:

Table 1. Performance of the model evaluated on MorphoRuEval-2017 test data

genre	accuracy by tokens	# tokens	# correct tokens	accuracy by sentences	# sentences	# correct sentences
fiction	97.45%	4,042	3,939	81.98%	394	323
news	97.37%	4,179	4,069	87.71%	358	314
social	96.52%	3,877	3,742	81.34%	568	462

The accuracy by sentences metric shows the fraction of sentences where each word was tagged correctly.

The degradation of performance on the social media texts should be the case of the genre differences between the train and test sets. Besides, the design of our algorithm leads to better performance on texts with proper spelling and good grammar. Frequent misspellings in the social media text limit the ability of the method to use the lexicon.

4.2. Errors Analysis

Table 3 shows the most frequent mistakes that our algorithm made during the MorphoRuEval-2017 competition. The "Number of occurrences" column shows frequency of the correct tag in the test selection, the "Number of error" column shows the number of cases when another tag was mistakenly predicted.

Table 3. Frequencies of the most common errors made by our system

Correct tag	Number of occurrences	Predicted tag	Number of errors
Nominative	2,650	Accusative	60
Accusative	1,644	Nominative	37
Plural	2,777	Singular	28
Nominative	2,650	Genitive	19
DET	656	PRON	14
PRON	1,133	DER	11

[5] From magazines.russ.ru

[6] From lenta.ru

[7] From vk.com

About 30% of all mistakes are the result of the ambiguity between nominative and accusative cases. The architecture of our network was designed to deal with such ambiguity by usage of the whole context of the word. However, even LSTM networks cannot perform well on long dependencies (which is a case of the gradient vanishing described in the 2.4 section). On the other hand, the system tends to follow the agreement between the tag of noun and its modifiers. Therefore, the incorrectly chosen tag of a noun usually leads to errors additional errors in the predicted grammatical value of the modifiers.

The mistakes in the determination of the number of nouns are also quite frequent—around 11% of all errors. For example, word "спортсменки" may be singular and in the genitive case ("sportswoman") or plural and in the nominative case ("sportswomen").

Another type of common errors is connected with distinguishing between some determiners and pronouns. Word "его" can have either "он" ("he") or "его" ("his") lemma and be either pronoun or determiner.

However, the fraction of such errors seems to be insignificantly low compared to the number of occurrences of the correct tags. The resolution of the ambiguity between the nominative and accusative cases seems to be the main issue of the algorithm.

4.3. Model Parameters Comparison

Using the test data provided by organisers we tested our model with different parameters.

Table 4 summarises the received results. The Accuracy columns contain the information about accuracies by tokens and by sentences.

Table 4. Comparison of performance of different models

Model	Fiction Accuracy	News Accuracy	Social Accuracy
Emb(5000)-1LSTM(768)-Dropout(0.2)-WithSuffixes	92.75% / 59.90%	94.52% / 55.59%	92.03% / 60.39%
Emb(5000)-1BiLSTM(768)-Dropout(0.2)-WithSuffixes	94.95% / 69.54%	97.01% / 75.70%	94.30% / 71.30%
Emb(5000)-1BiLSTM(768)-Dense(768)-Dropout(0.2)-WithSuffixes	95.35% / 71.83%	97.20% / 76.82%	94.66% / 73.94%
Emb(5000)-1BiLSTM(768)-2BiLSTM(512)-Dense(768)-Dropout(0.2)-WithoutSuffixes	**95.62%** / **74.11%**	97.37% / 77.65%	94.97% / 74.65%
Emb(5000)-1BiLSTM(768)-2BiLSTM(512)-Dense(768)-Dropout(0.2)-WithSuffixes	95.57% / 73.10%	97.37% / 78.77%	95.13% / 74.47%
Emb(5000)-1BiLSTM(768)-2BiLSTM(512)-Dense(768)-Dropout(0.5)-WithSuffixes	95.30% / 73.35%	**97.54%** / **79.89%**	**95.15%** / **75.00%**
Emb(50000)-1BiLSTM(768)-2BiLSTM(512)-Dense(768)-Dropout(0.2)-WithSuffixes	95.27% / 71.57%	97.03% / 76.54%	95.00% / 74.65%
Final Variant	97.45% / 81.98%	97.37% / 87.71%	96.52% / 81.34%

The names of models reflect the architecture of the used network. “Emb” parameter shows the number of words in the embeddings layer. The following parameters show the types of layers and numbers of neurones in them. The last parameter indicates whether the suffix features were incorporated.

All models except the last one were trained on GICR corpus only. The last (“Final variant”) refers to the model described in section 3.5.

The model with single LSTM layer shows the worst tagging quality. Obviously, the context information received from the left context only is not sufficient for proper tagging.

Clearly, the system gains from additional layers: the next two models with a single Bidirectional LSTM layer perform worse than more complicated models. On the other hand, larger embeddings layer (the Emb(50000)-model) also leads to poor accuracy. It can be explained by the lack of train data: we should use much bigger corpus to train such embeddings.

The increase in the dropout values helps to achieve a little better accuracy. Besides, the suffix features give an improvement in the model’s performance.

It should be noted that the model with single Bidirectional LSTM layer and only 5 thousand words in embeddings achieves good enough results in comparison with our final model while it is 2.5 times smaller. For some applications, such model could be more plausible than large but accurate one.

5. Conclusion

We have developed a POS-tagging model for Russian that can achieve high accuracy. Our system showed the best results on the MorphoRuEval-2017 competition. The degradation of its performance on some genres seems to be reasonably insignificant. Our model takes advantage of vast morphological description and modern machine learning techniques. Such approach seems likely to bring improvements in the quality of NLP-analysis systems based on the morphological analysis.

References

1. *Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A.* (2012), Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. Computational linguistics and intellectual technologies: Proceedings of the International Conference “Dialog 2012”. Vol. 2, pp. 91–103
2. *Bengio Y., Simard P., Frasconi P.* (1994), Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, Vol. 9, Issue 2, pp. 157–166.
3. *Choi J.* (2016), Dynamic Feature Induction: The Last Gist to the State-of-the-Art, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (NAACL 2016), San Diego, CA, pp. 271–281.
4. *Georgiev G., Zhikov V., Osenova P., Simov K., Nakov P.* (2012), Feature-rich part-of-speech tagging for morphologically complex languages: application to Bulgarian, Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, pp. 492–502.

5. *Giesbrecht E., Evert S.* (2009), Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus, Proceedings of the Fifth Web as Corpus Workshop (WAC5), pp. 27–35.
6. *Hochreiter S., Schmidhuber J.* (1997), Long Short-Term Memory, Neural Computation, Vol. 9, Issue 8, pp 1735–1780.
7. *Ioffe S., Szegedy Ch.* (2015), Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, available at: https://arxiv.org/abs/1502.03167.
8. *Kingma D., Ba J.* (2014), Adam: A Method for Stochastic Optimization, available at: https://arxiv.org/abs/1412.6980.
9. *Ma X., Hovy Ed.* (2016), End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, available at: https://arxiv.org/abs/1603.01354
10. *Schmid H.* (1994), Probabilistic Part-of-Speech Tagging Using Decision Trees. Proceedings of International Conference on New Methods in Language Processing, Manchester, UK.
11. *Segalovich I.* (2003), A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications, Las Vegas, Nevada, USA.
12. *Selegey D., Shavrina T., Selegey V., Sharoff S.* (2016) Automatic morphological tagging of Russian social media corpora: training and testing, Computational linguistics and intellectual technologies: Proceedings of the International Conference "Dialog 2016", pp. 589–604
13. *Sharoff S., Nivre J.* (2011), The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2011" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2011"], Bekasovo, pp. 591–605.
14. *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* (2014), Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, Vol. 15 Issue 1: pp. 1929–1958.
15. *Theano Development Team* (2016), Theano: A Python framework for fast computation of mathematical expressions, available at: https://arxiv.org/abs/1605.02688.
16. *Toutanova K., Klein D., Manning C., Singer Y.* (2003), Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252–259
17. *Zalizniak A.* (1977) Russian Grammar Dictionary [Grammaticheskii Slovar' Russkogo Iazyka. Russki Iazyk].

SEMANTIC DESCRIPTIONS FOR A TEXT UNDERSTANDING SYSTEM

Boguslavsky I. (bogus@iitp.ru)

Laboratory of Computational Linguistics, A. A. Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Russia

Departamento de Inteligencia Artificial, ETSII, Universidad Politécnica de Madrid, Spain

The semantic analyser SemETAP is a module of the ETAP-3 Linguistic Processor. It uses 2 static semantic resources—the combinatorial dictionary and the ontology. The former contains multifarious information about the words, and the latter stores extralinguistic (world) knowledge on the concepts and serves as the metalanguage for semantic description. World knowledge is needed, on the one hand, to enhance text analysis, and, on the other hand, to extract implicit information by means of inference. Both words and concepts are supplied with semantic descriptions. A semantic description consists of a definition in a formal language, which can optionally contain implications and expectations. For user's convenience, the description may also be provided by examples and a definition in NL. Semantic descriptions of several words and concepts are given.

Keywords: language model, ontology, deep text analysis, semantic definitions, implications, expectations

1. SemETAP Semantic Analyser

Semantic analyser, called SemETAP, is a module of the ETAP-3 multifunctional linguistic processor. Its goal is to provide semantic interpretation of texts using linguistic and world knowledge. Examples of how SemETAP operates can be found in Boguslavsky et al. 2015 and Boguslavsky 2016. Semantic descriptions of words and concepts are a key component of SemETAP. The content and the format of these descriptions are determined by the design and the goals of SemETAP. Its main features are as follows.

- Rule-based approach. The system is mostly knowledge-based, although some modules contain data-driven components.
- Stratification. Each sentence is represented by a series of structures, which correspond to various representation levels. These are (a) Morphological structure, (b) Syntactic structure, (c) Normalised syntactic structure, (d) Basic semantic structure, and (e) Enhanced semantic structure.
- Balance between the rules and the dictionary. Linguistic knowledge is distributed among two resource types—static (dictionaries and ontology) and dynamic (sets of rules), which interact strongly.

- Linguistic and world knowledge. As opposed to many other semantic processing systems, including advanced semantic parsers, such as StanfordCoreNLP[1], Boxer[2], WASP[3] and KRISP[4], SemETAP uses not only linguistic but also world knowledge. We share this approach with several other knowledge-oriented projects, which also rely on using detailed semantic and ontological information (cf. e.g. Pustejovsky 1991, Nirenburg and Raskin 2004, Mairal and Usón 2009, Anisimovich et al. 2012, Cimiano et al. 2014). Semantic descriptions created for SemETAP are distributed between the Combinatorial dictionary and the Ontology. Both resources use the same metalanguage, based on the ontological elements.
- Focus on inference. We assume that the level of text understanding achievable by the semantic analyser is determined by the amount of inferences the system can draw. Therefore, the major goal of the analyser is twofold: it should (a) construct the Basic Semantic Structure (BSemS) of each sentence, and (b) draw all possible inferences from BSemS, which results in the Enhanced Semantic Structure (EnSemS).

Among the inferences envisaged in semantic descriptions, we distinguish between strict logical entailments (implications) and plausible expectations. Both play an important role in interpreting coherent discourse and dialogues. An implication of an utterance is an inference that is necessarily true. For example, sentence *John broke the cup* necessarily implies that the cup has lost its integrity. A plausible expectation takes place when a certain state-of-affairs can be expected in the given situation but it is not obligatory. When somebody says that *John dropped the cup* we can reasonably expect that the cup will be broken, but we will easily accept the opposite. Sometimes, an utterance allows for both types of inference. For example, the literal meaning of the sentence *John went to the university (at moment t)* that constitutes its BSemS is that at *t* John began moving towards the university with the aim of being there. Out of this BSemS, one can draw two conclusions that differ in power. The first one has the status of a logical entailment and hence is completely true: "at *t* John ceased to be at the initial point of his movement". The second inference is merely a plausible expectation: "it can be expected that at some later moment *t1* John will be at the university". It can be shown that during the interpretation of discourse, plausible expectations play an even greater role than logical implications. Apparently, it is plausible expectations that mostly guarantee text coherence and help restore omitted pieces of information. For example, the sentence *Mother asked me to repair the fence* does not logically imply that the fence has been repaired. However, this is a plausible expectation entailed by the meaning of *asked.* Therefore, we perceive the following dialogue as coherent: *What were you doing yesterday?—Mother asked me to repair the fence.* Although the reply does not give a direct answer to the question, the hearer extracts the answer due to the expectation generated by *asked.* In compiling semantic descriptions, we tried to pay special attention both to implications and to plausible expectations.

[1] http://nlp.stanford.edu:8080/corenlp/

[2] http://www.let.rug.nl/bos/pubs/Bos2015NoDaLiDa.pdf

[3] http://www.cs.utexas.edu/~ml/wasp/

[4] http://www.cs.utexas.edu/~ml/krisp/

2. Ontology and Combinatorial dictionary

As mentioned above, the system disposes of two semantic resources—the Combinatorial dictionary and the Ontology. The Ontology plays a double role in the project. On the one hand, it serves as a structured source of world knowledge; on the other hand, ontological elements (concepts, instances, ontological relations) constitute a unique metalanguage of semantic description. This means that all sense-bearing text elements should be interpreted in ontological terms. This makes the task of establishing the links between the dictionary and the ontology far from trivial (Boguslavsky et al. 2010).

Combinatorial dictionary has a ramified structure and contains many types of information (Apresian et al. 2003). It is distributed among the following zones:

1 Lexeme name
2 Syntactic features
3 Semantic features
4 Government pattern (subcategorization frame)
5 Lexical functions
6 Zones of translation to another language—a separate zone for each working language (English, German, Spanish, Korean, UNL, Ontology-based semantic language)[5]
    6.1 Default equivalent
    6.2 Translation rules
7 Other types of rules operating at various stages of processing.

Dictionary entry zones relevant for this paper are the Semantic features zone (3), Government pattern (4) and the Semantic language zone (6).

Semantic features are used in all ETAP-3 options, including semantic analysis. They are referred to by the rules of various types, first of all by semantic agreement rules. Until recently we used a set of 57 features. Last year these features were replaced by ontology concepts, which allow a much more detailed representation of semantic properties of words. One of the consequences of this operation is that semantic restrictions in the government pattern can now be formulated in terms of ontological concepts and not in terms of semantic features used before.

If the word has valencies, its syntactic government pattern is supplemented by the semantic one, to show which element of BSemS corresponds to each syntactic actant.

Zone 6 of the entry gives a semantic equivalent of the word. If the word has a direct correspondence among the ontology concepts, it is given in zone 6.1 “Default equivalent”. The semantics of this concept is described in the ontology. If the ontology contains no direct equivalent of the word, and its introduction is not expedient for any reason, semantic description of the word is given in zone 6.2 by means of a full-fledged BSemS. The choice between these alternatives depends on various considerations on which we cannot dwell here. Still, this choice is often a matter of convenience, because both the words and the concepts are described in the same metalanguage and according to the same principles.

[5] ETAP-3 is a multilingual system, and the dictionary is designed so as to permit each word to be translated to several languages. For each translation language, the dictionary entry has a special zone. For this paper, only translation to the semantic language is relevant.

OntoEtap ontology is built on the basis of the popular and freely downloadable SUMO ontology (www.ontologyportal.org), which we supplemented by a considerable amount of data necessary for semantic analysis of natural languages (Boguslavsky 2011). An important property of all ontologies, which we actively exploit, is the top-down inheritance of all properties.

From the formal point of view, a semantic structure is a set of triples of the type `relation(Ontoelement-1,Ontoelement-2)`, where `relation` is an object or data property of the ontology, and `Ontoelement-i` is a variable or a constant denoting a concept or an instance. This rdf-formalism was chosen because, on the one hand, it is very flexible and expressive, and on the other hand, it is supported by a wide range of tools and is easily integrated with many Semantic Web applications.

3. Semantic descriptions of words and concepts

As mentioned above, semantic descriptions of words and concepts are carried out in the same semantic metalanguage and according to the same principles. As of now, we described a number of concepts and Russian words belonging to various semantic classes[6]: mental predicates (*want, patience, understand*), events (*ball, examination, interview*), instruments (*saw, axe, frying-pan*), animals (*dog*), plants and fruit (*apple, olive*), body parts (*hand, face, breast, pelvis*), time (*noon, midnight, soon*), subjective attributes (*cautious, dangerous, sympathetic, daring*), natural phenomena (*frost, hot weather, cool weather*), emotional states (*anger, grieve, resentment*), transport (*airplane, helicopter*), organizations (*restaurant, library*), and some other.

In completeness and detail, semantic descriptions are close to modern lexicographic definitions, but often surpass them in the amount of world knowledge. However, these descriptions do not replace encyclopedia. We include only such world knowledge that may be useful for commonsense reasoning—although clear boundaries are obviously very difficult to draw.

As an illustration, we provide a layout for the description of physical objects. In parentheses, we give corresponding ontological relations. Aspects to be taken into account while describing an Object include the following:

- parts of the Object; obligatory (hasPart): *bird—wing, house—roof;* or typical (hasTypicalPart): *house—attic, loft, cellar.*
- something that the Object is part of (inverse to hasPart and to hasTypicalPart—isPartOf, isTypicalPartOf): *window—building, transport*.
- typical size (height, weight …) of the Object (hasSize, hasHeight, hasWeight, …): *apple—10 cm*.
- typical material or an object the Object consists or is made of (isMadeOf): *book—paper, book cover—cardboard, fruit juice—fruit, porc—pig*.
- things that are typically made of the Object (inverse to isMadeOf—isMaterialFor): *fruit—juice, milk—cheese, timber—furniture, wood—furniture, gold—jewelry*.
- typical form of the Object (hasForm): *pill—round*.

[6] For simplicity, we represent examples with English words rather than concept names or Russian words.

- typical colour of the Object (hasColour): *apple—red, green, yellow*.
- typical location of the Object (hasTypicalLocation): *fish—in a natural body of water* or *in an aquarium, fruit—in the orchard, cloudberry—in the tundra*.
- typical origin (hasOrigin): *avocado—Southern region, camembert—France*.
- major predestination of the Object (hasFunction): *axe—chop* (as an instrument), *pen—write* (as an instrument), *food—eat* (as an object), *beverage—drink* (as an object). The predestination of *hen* for being eaten is accounted for by the fact that it is included not only in the class Poultry (which does not have any predestination) but also in the class Food (which does) and inherits hasFunction Eating (as an object) from this class.
- situations in which the Object frequently takes part, different from the main predestination (participatesIn): *axe—draw nails* (as an instrument), *knife—kill* (as an instrument), *hen—boil, fry, feed* (as an object), *lay eggs* (as the subject).

Some of these data, such as the typical location or frequent situations are placed in the section Expectations (see below), if the probability of their being true in all cases is not high enough.

In a general case, a semantic description contains the following sections:

1. Examples.
2. Definition or explanation in natural language.
3. Definition in a formal language, which may include Implications and Expectations.

The first two sections are intended for humans and written in natural language, and the third section is written in the formal language and used for semantic analysis and inference. From the formal point of view, the definition is a rule whose left part is the word (or a concept) and possibly a set of conditions, and the right part is a BSemS.

Below are several semantic descriptions of words and concepts of different classes supplied with detailed comments.

4. Examples of semantic descriptions

Below we will illustrate semantic descriptions by one word (*pomogat'* 'to help') and several concepts.

4.1. *Pomogat'* 'to help'

We will take the word *pomogat'* 'to help' in its major sense represented in examples (1)–(4). In square brackets are elements of BSemS (which will be explained below) corresponding to the actants of *pomogat'*.

<u>Examples:</u>

(1) *Kolja* [Agent1] *pomogaet Mashe* [Agent2] *reshat'* [Event2] *zadachu*.
'Kolja [Agent1] helps Masha [Agent2] solve [Event2] the problem'

(2) *Uchitel'* [Agent1] *pomogaet ucheniku* [Agent2] *v vybore* [Event2] *temy dlja sochinenija*.

‘the teacher [Agent1] helps the pupil [Agent2] to chose (lit. in the-choice of) [Event2] the topic for the composition’

(3) *On* [Agent1] *pomog mne* [Agent2] *s perevodom v Moskvu* [Event2] *i s zhiljem* [Object1].
‘he [Agent1] helped me [Agent2] with the transfer to Moscow [Event2] and with the lodging [Object1]’

(4) *On* [Agent1] *vsegda gotov pomoch den’gami* [Object2] *i sovetom* [Event4].
‘he [Agent1] is always willing to help with money [Object2] and advice [Event4]’.

NL definition:

“Agent1 has the goal of doing Event2 or obtaining Object1. Agent2 has the goal of facilitating this to Agent1. Therefore Agent2 is doing Event4 or is giving Object2 to Agent1. It is good for Agent1 that Agent2 is doing this”.

Formal definition:

To make the formal definition more illustrative, we will represent it by a commented table.

Pomogat’ →	
hasObject(?Goal1, ?Agent1)	?Agent1 has the goal of performing ?Event1,
hasObject2(?Goal1, ?Event1)	
hasAgent(?Event1,?Agent1)	
hasAlternative(?Event1, ?Event2)	which is either ?Event2 (*solve* in (1), *chose* in (2), *transfer* in (3))
hasAlternative(?Event1,?Getting)	or getting
hasObject(?Getting,?Object1)	?Object1 (*lodging* in (3))
hasObject(?Goal2,?Agent2)	?Agent2 has the goal of
hasObject2(?Goal2,?Facilitating)	facilitating
hasObject(?Facilitating,?Event1)	?Event1
hasBeneficiary(?Facilitating,?Agent1)	for ?Agent1
hasAgent(?Event3,?Agent2)	?Agent2 performs ?Event3
hasAlternative(?Event3,?Event4)	which is either ?Event4 (*advice* in (4))
hasAlternative(?Event3,?Giving)	or giving
hasObject(?Giving,?Object2)	?Object2
hasRecipient(?Giving,?Agent1)	to ?Agent1
hasObject(?EvalModality, ?Event3)	?Event3 is good for ?Agent1
hasBeneficiary(?EvalModality, ?Agent1)	
hasValue(?EvalModality, HighDegree)	

Implication: if *POMOGAT′* = past,perf, then Agent2 performed Event3 and Agent1 performed Event1.

Petr pomog Mashe reshit’ zadachu → *Masha reshila zadachu*
‘Petr helped Masha solve the problem’ → ‘Masha solved the problem’

Uchitel' pomog ucheniku v vybore temy → *Uchenik vybral temu*
'The teacher helped the pupil choose the topic' → 'The pupil chose the topic'

On pomog mne s perevodom v Moskvu → *Ja perevelsja v Moskvu*
'He helped me with the transfer to Moscow' → 'I transferred to Moscow'

On pomog mne s zhiljem → *Ja poluchil zhilje*
'He helped me with the lodging' → 'I got the lodging'

On pomog mne den'gami i sovetom → *On dal mne den'gi i sovet, i ja sdelal to, chto xotel sdelat'*
'He helped me with money and advice' → 'He gave me money and advice, and I did what I wanted to'

Note the last example: although the initial sentence does not mention the goal that Agent2 wishes to achieve, one can infer that the goal has been met.

Expectation: if *POMOGAT'* =nonpast or imperf, then it can be expected that: Agent2 performs Event3 and Agent1 performs Event1.

Petr pomogaet (pomozhet) Mashe reshit' zadachu → It can be expected that: *Masha reshit zadachu*.
'Petr helps (will help) Masha solve the problem' → It can be expected that: 'Masha will solve the problem'

Below are descriptions of concepts.

4.2. Apple

Example:

Eva sorvala s dereva jabloko i ugostila Adama
'Eva plucked an apple and gave it to Adam'

NL definition: "A fruit as big as a fist growing on apple tree, of round shape. Having a red, yellow or green colour, contains juicy flesh, peel, small brown seeds, good for health, of sweet or sour-sweet taste".

Formal definition:

Apple(?Apple) →	If there is an instance ?Apple of the Apple concept, then:
Fruit(?Apple)	it belongs to the Fruit class. The latter, in its turn, belongs to the Food class, whose predestination is being eaten. The description of Eating includes the proposition that the goal of eating is to satisfy hunger or to enjoy. All these data are inherited by Apple and other Fruit.

hasObject(?BeFruitOf, ?Apple)	Apple is a fruit of an apple tree.
hasObject2(?BeFriutOf, ?AppleTree)	
hasPart(?Apple, ?Thing1)	Here major parts of an apple are listed: - seeds, which are: small brown - stem - skin - juice
hasSubset (?Thing1, ?Seed)	
hasSize(?Seed, Small)	
hasColor(?Seed,Brown)	
hasSubset (?Thing1, ?Stem)	
hasSubset (?Thing1, ?Skin)	
hasSubset (?Thing1, ?Juice)	
hasObject(?HavingSize,?Apple)	Here the size of a typical apple is given. Since our descriptions are intended for commonsense reasoning, we prefer to describe the size of objects in absolute numbers, though approximate, and not by means of anthropomorphic reference ("size of a fist"), as it is done in lexicography. The typical size of an apple is about 10 centimetres.
hasValue(?HavingSize, ?LinearMeasure)	
inUnit(?LinearMeasure, Centimeter)	
hasNumericalValue(?LinearMeasure, 10)	
has Attribute(?Apple,?Attribute)	Apple has several attributes:
hasSubset(?Attribute, ?ColorAttribute)	- colour,
hasSetOrAlternative(?ColorAttribute, Red)	which can be red
hasSetOrAlternative(?ColorAttribute, Yellow)	yellow or
hasSetOrAlternative(?ColorAttribute, Green)	green hasSetOrAlternative relation denotes non-exclusive disjunction, as opposed to hasAlternative, which corresponds to exclusive disjunction
hasSubset(?Attribute, Round)	- round shape
hasSubset(?Attribute,?TasteAttribute)	- taste, which can be
hasAlternative(?TasteAttribute, Sweet)	sweet or
hasAlternative(?TasteAttribute, Sour-sweet)	sour-sweet
hasSubset(?Attribute, GoodForHealth)	- good for health
hasSubset(?Attribute, Juicy)	- juicy
hasSubset(?Attribute, Crisp)	- crisp

Expectations:

participatesIn(?Apple,?Eating)	Typical situations in which apples participate:
hasObject(?Eating,?Apple)	- Eating (as an object) (inherited from Food)
participatesIn(?Apple, ?Baking)	- Baking (as an object)
hasObject(?Baking, ?Apple)	
participatesIn(?Apple, ?Squeezing)	- squeezing apple juice
hasObject(?Squeezing, ?Apple)	
hasResult(?Squeezing, ?AppleJuice)	
participatesIn (?Apple, ?Making)	- Making such objects as:
hasObject(?Making, ?Thing2)	
hasSubset(?Thing2, ?ApplePie)	ApplePie
hasSubset (?Thing2, ?AppleJam)	AppleJam
hasSubset (?Thing2, ?Cider)	Cider
isMaterialFor(?Apple, ?Thing2)	Apple participates in the manufacturing of these objects as an ingredient
hasTypicalLocationAt(?Apple, ?Thing3)	Typical places where one can find Apple:
hasAlternative(?Thing3, ?Orchard)	- Orchard
hasAlternative (?Thing3, ?AppleTree)	- AppleTree
hasAlternative (?Thing3, ?GroceryStore)	- GroceryStore
hasAlternative (?Thing3, ?House)	- House
hasAlternative(?Thing3, ?Bowl)	- Bowl
hasAlternative (?Thing3, ?Fridge)	- Fridge

4.3. Heating

Example: *Nagrevaem smes' do kipenija, a potom oxlazhdaem* 'we heat the mixture until it boils and then cool it'. *Prodavcy tropicheskix rybok obogrevali akvariumy kerosinovymi lampami* 'the sellers of tropical fish heated aquariums with oil lamps'.

NL definition: "The temperature of ?Object increases from ?Quant1 to ?Quant2"

Formal definition:

Heating(?Heating) →	if there is an instance ?Heating of Heating, then:
IncreasingProcess(?Heating)	it belongs to the IncreasingProcess class
hasObject(?Heating, ?Object)	there is an Object that undergoes this process
hasTime(?Heating, ?TimeInterval)	over the time interval ?TimeInterval
begins(?Time1, ?TimeInterval)	?Time1 is the beginning of ?TimeInterval
ends(?Time2, ?TimeInterval)	?Time2 is the end of ?TimeInterval

hasObject(?HavingTemperature1, ?Object)	at ?Time1 ?Object's temperature is equal to ?Quant1
hasValue(?HavingTemperature1, ?Quant1)	
TemperatureMeasure(?Quant1)	
hasTime(?HavingTemperature1, ?Time1)	
hasObject(?HavingTemperature2, ?Object)	at ?Time2 ?Object's temperature is equal to ?Quant2
hasValue(?HavingTemperature2, ?Quant2)	
TemperatureMeasure(?Quant2)	
hasTime(?HavingTemperature2, ?Time2)	
greaterThan(?Quant2, ?Quant1)	?Quant2 is greater than ?Quant1

4.4. HeatingDevice

Example: *Nagrevatel'noe ustrojstvo USP-2 prednaznacheno dlja podogreva plastin na raznyx stadijax analiza* 'the heating device USP-2 is intended for heating plates at various stages of the analysis'.

NL definition: "A device that serves as an instrument of heating something, e.g. electric heaters, heat lamps, ovens, stoves, etc."

Formal definition:

HeatingDevice(?HeatingDevice) →	if there is an instance ?HeatingDevice of HeatingDevice, then:
Device(?HeatingDevice)	?HeatingDevice belongs to the ?Device class
hasFunction(?HeatingDevice, ?Heating)	The function of ?HeatingDevice consists in serving as an instrument in the ?Heating process
hasInstrument(?Heating, ?HeatingDevice)	

4.5. Stove

Example: *Nekotorye pechi rabotajut na neetilirovannom benzine* 'some stoves are fuelled with unleaded petrol'.

NL definition: "A device used for heating a room or for cooking, which works by burning wood, coal, oil, petrol or gas or is powered by electricity".

Formal definition:

Stove(?Stove) →	If there is an instance ?Stove of the Stove concept, then:
HeatingDevice(?Stove)	it belongs to the HeatingDevice class
hasFunction(?Stove, ?Heating1)	serving for heating is inherited from HeatingDevice (cf. above)
hasInstrument(?Heating1, ?Stove)	
hasGoal(?Heating1, ?Event1)	in the Stove, the heating is made either for

hasSubsetOrAlternative(?Event1, ?Heating2)	heating buildings or parts thereof (StationaryArtifact)
hasObject(?Heating2, ?StationaryArtifact)	
hasSubsetOrAlternative(?Event1, ?Cooking)	or for cooking, or for both
isResultOf(?Heating, ?Event2)	Heating is obtained either by
hasAlternative(?Event2,?Burning)	- burning wood or coal or oil or petrol or gas
hasObject(?Burning,?Substance)	
hasAlternative(?Substance, ?Wood)	
hasAlternative(?Substance,?Coal)	
hasAlternative(?Substance,?Oil)	
hasAlternative(?Substance,?Petrol)	
hasAlternative(?Substance,?Gas)	
hasAlternative(?Event2,?Using)	- or by using electricity
hasObject(?Using,?Electricity)	

4.6. Organization

Example: *Many international organizations have their headquarters in Geneva.*

NL definition: "Group of people whose activity is coordinated to attain common goals".

Formal definition:

Organization(?Organization) →	
Group(?Organization)	Organization belongs to two classes—Group
Agent(?Organization)	and Agent
hasChief(?Organization,?Human1)	Organization has a chief
hasInStaff(?Organization,?Human2)	and staff
hasFunction(?Organization,?Action)	Organization has a primary function—to do something

4.7. ClientServingOrganization

NL definition: "Organization whose function is to provide services to clients".

Formal definition:

ClientServingOrganization (?CS-Organization) →	
Organization(?CS-Organization)	CS-Organization belongs to Organization and inherits all its properties
hasUser(?CS-Organization,?Agent)	CS-Organization has users that may be people or organizations
hasSubset(?Agent,?Human)	
hasSubset(?Agent, ?Organization)	

hasUserAction(?CS-Organization,?Action)	there is a typical action that a user of the CS-Organization performs. E.g. in a shop it is buying things, in a hospital it is receiving treatment and in a movie theatre it is watching a film.

4.8. Library

NL definition: “Organization that has a collection of sources of information and similar resources, and makes them accessible to clients”

Formal definition:

Library(?Library)	
ClientServingOrganization(?Library)	Library is a subclass of ClientServingOrganization and inherits all its properties (which we do not repeat here)
belongsTo(?Library, ?PhysicalObject1)	Library belongs to Organization or Region. The belongsTo slot is often filled in Russian by adjectives or genitive noun phrases: *Rajonnaja* ‘regional’, *gorodskaja* ‘city’, *shkol’naja* ‘school’, *sinodal’naja* ‘synodal’, *tjuremnaja* ‘prison’, *Administracii prezidenta* ‘President’s Administration’, *Akademii nauk* ‘Academy of Sciences’, *zavodskaja* ‘factory’, *oblastnaja* ‘provincial’, *kraevaja* ‘territorial’, *kafedral’naja* ‘departmental’, *korolevskaja* ‘royal’, *nacional’naja* ‘national’, *polkovaja* ‘regiment’, *universitetskaja* ‘university’
hasSubset(?PhysicalObject1, ?Organization)	
hasSubset(?PhysicalObject1, ?Region)	
hasUser(?Library, ?Human)	Users of libraries are also often expressed in Russian by adjectives: *kursantskaja* ‘for cadets’, *oficerskaja* ‘for officers’, *detskaja* ‘for children’, *obschedostupnaja* ‘public’, *rabochaja* ‘for workers’
hasFunction(?Library,?Lending)	The function of the library consists in lending ContentBearingObjects to its users.
hasObject(?Lending1,?ContentBearing Object)	
hasAddressee(?Lending1,?Human)	
hasInStock(?Library,?ContentBearing Object)	Library disposes of ContentBearingObjects of different kinds: books, journals, newspapers, audios, videos, maps, patents, etc.

hasUserAction(?Library,?Intentional PsychologicalProcess)	What users are doing is reading listening to or watching these ContentBearingObjects
hasSubsetOrAlternative(?Intentional PsychologicalProcess,?Reading)	
hasSubsetOrAlternative(?Intentional PsychologicalProcess,?Listening)	
hasSubsetOrAlternative(?Intentional PsychologicalProcess,?Watching)	
hasAgent(?IntentionalPsychological Process, Human)	
hasObject(?IntentionalPsychological Process,?ContentBearingObject)	
hasLocation(?Library,?PhysicalObject2)	?Library may be located in an ?Organization or in a ?Region: *moskovskaja* 'Moscow', *domashnjaja* 'home'
hasSubset(?PhysicalObject2, ?Organization)	
hasSubset(?PhysicalObject2,?Region)	
hasAboutness(?Library,?Entity)	?Library may cover a definite topic or domain, which is often expressed by adjectives: *istoricheskaja* 'historical', *medicinskaja* 'medical', *muzykal'naja* 'musical', *pedagogicheskaja* 'pedagogical', *politexnicheskaja* 'politechnical', *spravochnaja* 'reference', *teatral'naja* 'theater', *po obschestvennym naukam* 'social sciences', *estestvennyx nauk* 'natural science', *nauchnoj fantastiki* 'science fiction'.
hasTypicalPart(?Library, ?ReadingHall)	Library often has ReadingHall

Implications:

- hasAboutness(?Library,?Z) → hasInStock(?Library,?Z1)&hasAboutness(?Z1,?Z)) (If Library covers subject domain Z, then ContentBearingObjects it contains have topic Z, i.e. a historic library contains books on history)
- hasLocation(?Library,?Z)&Organization(?Z) → hasUser(?Library,?Z1)&(hasIn Staff(?Z,?Z1)/hasUser(?Z,?Z1)) (If Library is located in Organization, its users are either clients of this Organization or its employees, i.e. users of a university library are either students or university employees)

5. Conclusion

Our approach to semantic analysis lies within the knowledge-based paradigm. We are guided by the conviction that using explicit and detailed knowledge on the language and on the subject domain can be beneficial for many tasks. The compilation of detailed semantic descriptions, which include both linguistic and extralinguistic knowledge, is important in different perspectives. On the one hand, they are needed for modeling language competence, in the direction of both understanding and generation. It is no accident that encyclopedic knowledge was included in some entries of the theoretically oriented Explanatory-combinatorial dictionary of Russian (Mel'chuk et al. 1984). On the other hand, many semantically-aware applications, including word sense disambiguation, semantic parsing, question-answering, textual entailment, etc. may also benefit from the availability of this information. Its potential is even stronger when we think about such knowledge-intensive tasks as commonsense reasoning, implicit knowledge extraction or bridging anaphora.

Of course, we are aware of the fact that creation of such resources for the language at large or even for its large fragment is extremely time- and effort-consuming. We would certainly prefer obtaining the information needed by some data-driven technique. However, ontological and semantic information extracted nowadays automatically out of large volumes of data is less than adequate for the tasks we are facing. We do not see any immediate prospect of automating this process and prefer to carry it out to the best of our abilities by the means we dispose of now. If future researchers find ways of automatically extracting such (or similar) information out of data, our resource may serve as the baseline.

We believe that onto-semantic descriptions of the type proposed in this paper are a useful step towards accumulating formalized knowledge. Our future efforts will be directed towards enlarging the stock of these descriptions and testing them in different applications.

Acknowledgements

The development of semantic resources described in this paper is a joint work being carried out by a team of lexicographers at the Institute for Information Transmission Problems, that consists of V. Dikonov, T. Frolova, E. Inshakova, L. Iomdin, A. Lazursky, I. Rygaev and S. Timoshenko. The format of descriptions presented here, as well as many concrete decisions were discussed at our weekly seminars. I am happy to thank all these colleagues for their active participation, constructive criticisms and excellent proposals.

This work was supported by the RSF grant 16-18-10422, which is gratefully acknowledged.

References

1. *Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A.* (2012) Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference “Dialogue”, issue 11(18). Moscow. RGGU Publishers. p. 62–79
2. *Apresian Ju., Boguslavsky I., Iomdin L., Lazursky A., Sannikov V., Sizov V., Tsinman L.* (2003) ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT. First International Conference on Meaning-Text Theory (MTT'2003). June 16–18, 2003. Paris: Ecole Normale Superieure, p. 279–288.

3. *Boguslavsky I., L. Iomdin, V. Sizov, S. Timoshenko.* (2010) Interfacing the Lexicon and the Ontology in a Semantic Analyzer. In: COLING 2010. Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), Beijing, August 2010, pages 67–76.
4. *Boguslavsky I.* (2011) Semantic Analysis based on linguistic and ontological resources. Proceedings of the 5th International Conference on the Meaning—Text Theory. Barcelona, September 8–9, 2011. Igor Boguslavsky and Leo Wanner (Eds.), p. 25–36.
5. *Boguslavsky et al.* (2015)—Boguslavsky I., Dikonov V., Iomdin L., Lazursky A., Sizov V., Timoshenko S. Semantic Analysis and Question Answering: a System Under Development. In: Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" (2015), p. 62–79.
6. *Boguslavsky I.* (2016) On the Non-canonical Valency Filling. GramLex 2016, co-located with COLING 2016. Proceedings of the Workshop on Grammar and Lexicon: Interactions and Interfaces, pages 51–60, Osaka, Japan, December 11.
7. *Cimiano Ph., Unger Ch., McRae J.* (2014). Ontology-based Interpretation of Natural Language. Synthesis Lectures on Human Language Technologies #24. Morgan & Claypool Publishers.
8. *Mairal Usón, R. y J. C. Periñán-Pascual.* (2009). The anatomy of the lexicon component within the framework of a conceptual knowledge base. Revista Española de Lingüística Aplicada 22, 217–244.
9. *Mel'chuk I., Zholkovsky A.* (1984) Explanatory-combinatorial Dictionary of Modern Russian. [Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka]. Wien.
10. *Nirenburg, S., and V. Raskin.* (2004). Ontological Semantics. The MIT Press. Cambridge, Massachusetts. London, England.
11. *Pustejovsky J.* (1991). The Generative Lexicon. Computational Linguistics, vol. 17, N4, p. 409–441.

WHICH IR MODEL HAS A BETTER SENSE OF HUMOR? SEARCH OVER A LARGE COLLECTION OF JOKES

Bolotova V. V. (lurunchik@gmail.com),
Blinov V. A. (vladislav.blinov@urfu.ru),
Mishchenko K. I. (ki.mishchenko@gmail.com),
Braslavski P. I. (pbras@yandex.ru)

Ural Federal University, Yekaterinburg, Russia

This paper describes experiments on humorous response generation for short text conversations. Firstly, we compiled a collection of 63,000 jokes from online social networks (VK and Twitter). Secondly, we implemented several context-aware joke retrieval models: BM25 as a baseline, query term reweighting, word2vec-based model, and learning-to-rank approach with multiple features. Finally, we evaluated these models in two ways: on the community question answering platform Otvety@Mail.ru and in laboratory settings. Evaluation shows that an information retrieval approach to humorous response generation yields satisfactory performance.

Key words: computational humor, dialog systems, information retrieval approach, natural language processing

У КАКОЙ МОДЕЛИ ИНФОРМАЦИОННОГО ПОИСКА ЛУЧШЕ ЧУВСТВО ЮМОРА? ПОИСК В БОЛЬШОЙ КОЛЛЕКЦИИ ШУТОК

Болотова В. В. (lurunchik@gmail.com),
Блинов В. А. (vladislav.blinov@urfu.ru),
Мищенко К. И. (ki.mishchenko@gmail.com),
Браславский П. И. (pbras@yandex.ru)

Уральский федеральный университет,
Екатеринбург, Россия

1. Introduction

Following recent trends in the widespread use of dialog systems like Apple Siri, Microsoft Cortana, Google Now and others, it becomes important to incorporate sense of humor into them. Humorous responses can help to deal with out-of-domain queries which have become an issue for the chatbots. Moreover, jokes that occasionally appear during interaction can make appear dialog systems more human-like.

Sense of humor plays a significant role in human-computer interaction. In particular, (Nijholt, 2002; Khooshabeh et al., 2011) have shown that adding humor capabilities to conversational agents results in more trustable and attractive interaction for users. Furthermore, Nijholt (2002) has summarized research according to which a sense of humor is generally considered a valued characteristic of others and plays a significant role in some task-oriented interactions, e.g. teaching.

The aim of our study is to examine the effectiveness of information retrieval approach to humorous response generation. Firstly, we compiled a collection of 63,000 jokes from online social networks (VK and Twitter). Secondly, we implemented several context-aware joke retrieval models: BM25 as a baseline, query-term reweighting, word2vec-based, IBM model 1, and learning-to-rank approach with multiple features. Finally, we evaluated these model in two ways: on the community question answering platform Otvety@Mail.ru and in laboratory settings.

2. Related Work

There are two main research directions in computational humor: humor recognition and humor generation. Stock and Strapparava (2003) have considered the problem of generating funny expansions for known and unknown acronyms. For known acronyms the implemented system keeps some words unchanged (usually nouns) and finds contrasting but similarly sounding words for the remaining ones using WordNet and other linguistic resources. For unknown acronyms the system starts with a WordNet synset and generates a syntactically consistent but semantically incongruous sequence of words. Ritchie (2005) has systematized different types of puns and proposed mechanisms for automatic pun generation. Valitutti et al. (2013) have proposed a method how to make 'adult' puns from short text messages by lexical replacement. A related study (Hong and Ong, 2009) addresses the task of automatic template extraction for pun generation. The extracted templates consist of a syntax structure and binary relations between words (such as *SynonymOf*, *Compound-word*, *SoundsLike*, etc.). After the learning stage the authors obtained 27 templates. Best automatically generated jokes received about the same evaluation scores as the human ones.

The study (Mihalcea and Strapparava, 2006) proposes a method for adding a joke to an email message or a lecture note and is close to our approach. The solution exploits an automatically gathered collection of 16,000 one-liners. For a given text fragment the application finds the semantically closest joke using the latent semantic analysis (LSA). A small-scale users study showed good performance and reception of the proposed solution, though even returning a random joke provided relatively good performance (as an opposite to not adding any joke at all).

Yang et al. (2015) have drawn attention to *humor anchors*, i.e. words prompting comic effect, and have addressed the task of *humor anchor recognition*.

In the field of information retrieval, Friedland and Allan (2008) proposed a domain-specific joke retrieval model based on jokes structure and interchangeable word classes. Surdeanu et al. (2011) investigated usefulness of different linguistic features for search in large archives of questions and answers for non-factoid questions. The study does not deal with humorous content, but the approach is still similar to ours.

Ritter et al. (2011) studied the applicability of a data-driven approach for generating responses to Twitter status posts. They used phrase-based statistical machine translation while trying to solve the problem.

In our initial experiments (Blinov, 2016) we evaluated popularity-based ranking (Likes model). This model can be regarded as an analogue of query-independent ranking based on document authority (e.g. PageRank)—a funny joke is potentially still funny, even if it is not quite in the context. The model requires only minimal overlap between a question and candidate responses (one common noun or verb) and ranks the responses by descending normalized Like scores. However, evaluation showed that BM25 scoring outperforms simple joke popularity.

3. Data

3.1. Joke Collection

We gathered a collection of jokes from popular humor-related user communities and accounts on VK[1], the largest Russian online social network, and Twitter[2]. We collected posts without media content (images and video) that gained more than 500 "likes" for VK and at least 1 for Twitter (where "likes" are much rarer). The VK posts longer than 250 characters were eradicated. Table 1 summarizes the sources of the initial corpus.

Table 1. Initial collection of jokes by source

Community/Account	URL	Size
F*** Normality	https://vk.com/trahninormalnost1	70,647
Evil Incorporated	https://vk.com/evil_incorparate	69,431
Witty	https://vk.com/ostroym	42,267
Strange Humor	https://vk.com/c.umor	44,287
Humor FM	https://twitter.com/_humor_fm_	3,578
About Humor	https://twitter.com/abouthumor	332
Drunken Twitter	https://twitter.com/drunktwi	15,335
Caucasian Humor	https://twitter.com/kavhum	4,988
Funny Radio	https://twitter.com/veseloeradio	12,312
Jokes and Anecdotes	https://twitter.com/anecdot_eshe	5,181
	Total	268,358

Out of those posts we retained only one-liners and two-turn dialog jokes (see *Examples 1* and *2*, respectively), 226,431 jokes total. Then, we removed duplicates based on similarity of lemmatized bag-of-words representations. This step reduced collection size drastically—down to 63,293 jokes.

[1] https://vk.com/

[2] https://twitter.com/

Example 1:

Лекарства так подорожали, что скоро мы будем дарить их друг другу на день рождения... Чтобы дожить до следующего......))))))

Drugs have become so expensive that we will soon present them like a birthday gift... To attain the next anniversary......))))))

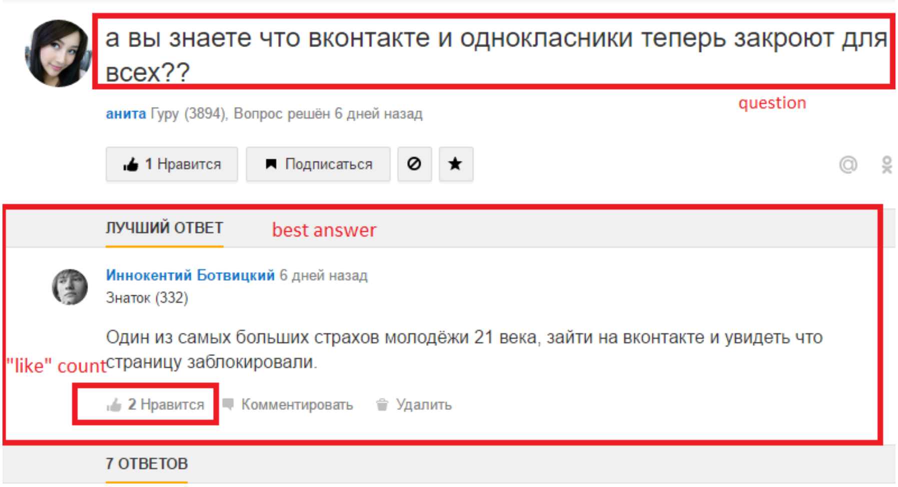
Example 2:

— Ты спать собираешься?
— Да, сейчас, закончу делать ничего и пойду.

— Are you going to go to bed?
— Yes, now, I finish doing nothing and go.

3.2. CQA Dataset

We also collected a large historical dataset of question-answer pairs from the Humor category of a popular Russian community question answering platform Otvety@Mail.Ru[3]. Each question there can be answered once by any user and each answer can be rated once by any user (Fig. 1 shows the user interface of the CQA platform). In addition, the asker can mark an answer as "the best" and then the question will be tagged as "solved". The collection that we compiled consists of more than 35,000 questions and more than 200,000 answers.



а вы знаете что вконтакте и однокласники теперь закроют для всех??

анита Гуру (3894), Вопрос решён 6 дней назад

Иннокентий Ботвицкий 6 дней назад
Знаток (332)

Один из самых больших страхов молодёжи 21 века, зайти на вконтакте и увидеть что страницу заблокировали.

Fig. 1. Otvety@Mail.ru interface

[3] https://otvet.mail.ru/humor

4. Retrieval Models

As a baseline model we chose BM25 (Jones et. al, 2000) scoring, which is based on textual similarity between queries and documents. Stimuli in this model are mapped to lemmatized bag-of-words representations without stop words and then are queried against an inverted index.

One drawback BM25 has is that it requires word overlap between a query and a response, while some relevant responses may have no common words with the query. In the study we propose two models that address this issue: a word2vec-based semantic similarity model and a learning-to-rank approach using a diverse set of features. We also propose a Query Term Reweighting model, which is an enhancement of BM25 scoring.

4.1. Query Term Reweighting (QTR)

The proposed approach follows the idea of "humor anchors" introduced in (Yang, 2015). "Humor anchors" are words and phrases that are important for comic effect. Constituents of "humor anchor" may have low *idf* weights. For instance, the response presented in *Example 3* will not be ranked high enough by the baseline model because pronouns have low *idf* across the corpus, while the approach described below picks this as its top-1 response.

Example 3:

Question: Я прекрасно знаю, как с тобой разговаривать, не учи меня!
QTR: Ты разговариваешь со мной так, как будто у тебя есть абонемент в больницу
BM25: Разговаривать с единорогами.

Question: I know how to talk to you perfectly well, do not teach me!
QTR: You talk to me as if you have a seasonal ticket to a medical center
BM25: To talk with unicorns.

Firstly, we processed dialog jokes from the joke collection using a lemmatizer[4] (Korobov, 2015). To figure out what kinds of words are important for comic effect, we analyzed which morphological tags appear frequently in both questions and corresponding answers. In particular, we used a combination of part of speech and grammatical case. The most popular tags, without considering prepositions and conjunctions, were nominal pronouns, nouns in the nominative case, and verbs. Based on the acquired data, we composed a set of rules described below to adjust weights of anchor words using empirically derived boosting weights (see Table 2). These rules were applied to every stimulus before using BM25 weighting. All non-anchor words were excluded, and *tf-idf* weights of anchor words were multiplied by the corresponding boost values.

1. **Subjects.** Since there is a lack of accurate syntactic parsers for Russian, we defined a subject simply as a noun or a pronoun in the nominative case: a person, a place, a thing, or an idea that acts or is being described in a sentence ("Mother" in *Example 4*). The subject was appended to a query with the highest boost.

[4] https://github.com/kmike/pymorphy2#citing

Example 4:
Мама накричала на папу.
Mother shouted at dad.

2. **Named entities.** Words marked as proper names were also considered as main anchor words ("Russia" in *Example 5*). These words were added to the query with the same boost as subjects.

Example 5:
Как мы можем обустроить Россию?
How can we develop Russia?

3. **Question word context.** All nouns that were within three-word window with interrogative words (e.g. "who", "which", "when", etc.), like "alcohol" in *Example 6*, were added to the query with the highest boost.

Example 6:
Как исключить алкоголь?
How to give up alcohol?

4. **Anchor word context.** We extended the query with adjectives that were grammatically related to the subject ("best" in *Example 7*), as well as objects in a three-word window with the subject ("dad" in *Example 4*). An object is a noun, a noun phrase, or a pronoun that is affected by the action of a verb (a direct object or an indirect object) or that completes the meaning of a preposition (the object of a preposition).

Example 7:
Кто лучший тренер?
Who is the best coach?

5. **Verbs.** When the subject was found in a stimulus, we added verbs with a boost lesser than the boost of objects. Otherwise, verbs were appended with the highest boost. For instance, in *Example 8* the query will be extended by words "do", "get" and "pregnant".

Example 8:
— Что делать, чтобы не забеременеть?
— Моя девушка спит с другими парнями, чтобы не забеременеть от меня.

— What to do to not get pregnant?
— My girlfriend sleeps with other guys to not get pregnant by me.

6. **Pronouns.** For every first person or second person pronoun in the stimulus, we appended to the query an "opposite by person" pronoun with the highest boost. For instance, for the pronoun "I" the opposite one is "you", for "our"—"their", and so on. The original pronoun was appended to the query with a lesser boost. In *Example 3* the pronouns "I" and "you" and in *Example 9* "your" and "my" will be added to the query.

Example 9:

— Какое твое любимое блюдо?
— Мое любимое блюдо — макароны с сыром, потому что их название содержит рецепт и список ингредиентов одновременно.

— What is your favorite dish?
— My favorite dish is pasta with cheese, because its name contains a recipe and a list of ingredients at the same time.

Table 2. Empirically derived anchor boosts

Anchor Type	Boost
Subject	4.0
Named entity	4.0
Question word context	4.0
Inflected pronoun	4.0
Verb (no subject)	4.0
Anchor word context	3.0
Verb	2.5
Pronoun	1.5

4.2. Word2vec-Based Document Embeddings

The word2vec (Mikolov et al., 2013) method is a way to obtain word vectors such that semantically similar words have close vectors in terms of cosine similarity. There are also techniques to obtain document vectors of the same kind. One of them is doc2vec (Le and Mikolov, 2014)—a method that can infer vectors for new documents after training on thousands of sample documents. However, given a word2vec model, we can find a document vector just by the sum of vectors for the document words. (Lau and Baldwin, 2016) suggests that even though the sum-based representation is less effective than doc2vec, it often has better performance than bag-of-words (n-grams) in semantic-based tasks. Considering the lack of a publicly available doc2vec model for Russian and the comparable performance of the sum-based approach, we used the latter to obtain document vectors.

Specifically, we used a word2vec model trained on a Russian news corpus and provided by the service RusVectōrēs[5] (Kutuzov and Kuzmenko, 2017). We followed the same preprocessing as during the word2vec model construction—each text was mapped to a list of units in the form “lemma_POS” by Yandex Mystem 3.0[6] analyzer. We precalculated document vectors for our joke collection, and then, given a stimulus, we calculated its vector and found the closest jokes in terms of cosine similarity between vectors.

[5] http://rusvectores.org/en/about

[6] https://tech.yandex.ru/mystem/

4.3. Learning-to-Rank (LETOR)

Analogously to (Surdeanu et al., 2011), we used a learning-to-rank algorithm with a diverse set of features to re-rank responses of other models. In particular, we built a pool of answer candidates using top-50 answers returned by the BM25, QTR, and word2vec-based models described above. We used RankLib implementation of RankBoost algorithm to obtain a ranking function. The algorithm was trained on the CQA dataset, employing the following features for a question-answer pair.

1. Question length in characters.
2. Answer length in characters.
3. Question length in tokens.
4. Answer length in tokens.
5. BM25 score for the question-answer pair. As the score value is not bounded, we normalized it using score for the top-ranked document, hence obtaining a value between 0 and 1.
6. QTR model score for the question-answer pair. This score was normalized in the same fashion as the BM25 score.
7. word2vec-based model score for the question-answer pair.
8. IBM Model 1 probability. The IBM model 1 infers a word translation probability table from a parallel corpus. This table can then be used to estimate the probability of the answer being a translation of the question (Brown et al., 1993), which is known to perform well as a feature in question-answering ranking (Surdeanu et al., 2011). We trained this model on the CQA dataset, and then applied the same empirical trick as in (Surdeanu et al., 2011): probability of a word translating to itself was set to 0.5, and all other translation probabilities for the word were re-scaled to sum to 0.5.
9. Presence of an imperative verb. This is a binary feature that indicates whether for any verb in the question the same verb is present in the answer, but in the imperative mood.
10. Number of nouns, verbs, and adjectives in the answer that do not appear in the question. This feature is referred to as the “informativeness” of the answer (Surdeanu et al., 2011).
11. Similarity of POS-tag sequences of the question and the answer. Tags were obtained via pymorphy2[7] library, and similarity was calculated using the “gestalt pattern matching” algorithm (Ratcliff and Metzener, 1988).
12. Presence of rhyming words. This is a way to capture “puns” in the answers. To detect rhymes, we used Metaphone (Binstock and Rex, 1995) algorithm, specifically an implementation for Russian language—MetaphoneRU[8] library. Originally, the algorithm is used to find similar-sounding last names, but we used it to match nearly-rhyming words.

BM25, QTR, word2vec and translation probability scores, as expected, gave the highest ranking performance impact.

[7] https://github.com/kmike/pymorphy2

[8] https://github.com/Reaverart/MetaphoneRU

5. Evaluation

We evaluated the models in two ways: in the Humor[9] category of the Otvety@Mail.ru CQA platform and in laboratory settings.

5.1. CQA Platform

Perhaps the most important distinguishing feature of this evaluation method in comparison with other methods is that there are considerably more users. In average, there are about two new questions posted each minute in the Humor category of Otvety@Mail.ru.

We automatically posted top-1 ranked responses of each model for randomly sampled questions from this category during four days and gathered user reactions after a week. In total, bots answered 267 questions due to strict limitations of the CQA platform for user actions (30 answers per day for new account).

Table 3 provides the results obtained from Otvety@Mail.ru.

Table 3. User reactions from Otvet@Mail.ru (267 questions)

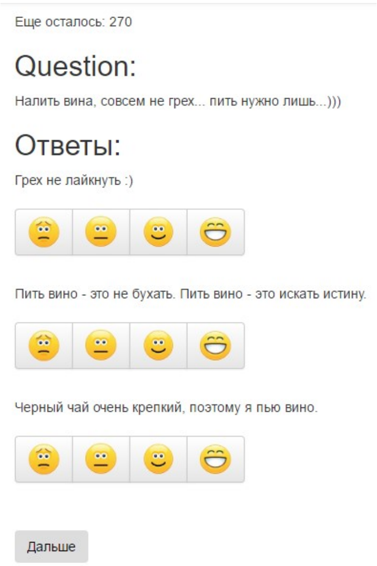
Model	Likes	# of best answers	Users that earned less likes	Best model
BM25	148	8	15.47%	14
QTR	142	16	14.91%	20
word2vec	147	14	15.82%	19
LETOR	156	12	16.93%	23
Oracle	197	50	24.67%	—

The "likes" column provides the total amount of "likes" for all answers of a model. The "best answer" column shows how many answers of the model were chosen as the best by question authors. The average percent of users who got less "likes" than the model in the same question thread is presented in the fourth column. Finally, the last column summarises how many times the model was better than the other ones. The model was considered as the "best" for a question if its answer was nominated as the "best answer" or earned more "likes" than answers of all other models. The last row of the table presents an "Oracle" model which chooses the most relevant answer within all mentioned models.

5.2. Lab Evaluation

Lab evaluation was conducted with the help of a dedicated annotation tool, see Fig. 2. Top-3 results for each model were selected for evaluation. Responses of all models were presented to an assessor in random order, three at a time. Responses were judged on a four-point scale (from 0 to 3, with the corresponding emoticons in the evaluation interface). We used pooling, each model was evaluated by four assessors independently. The assessors were instructed to pay close attention to question context during evaluation of the responses. As the test stimuli, we selected 80 questions from the ones we answered on the CQA platform.

[9] https://otvet.mail.ru/humor



Еще осталось: 270

Question:

Налить вина, совсем не грех... пить нужно лишь...)))

Ответы:

Грех не лайкнуть :)

Пить вино - это не бухать. Пить вино - это искать истину.

Черный чай очень крепкий, поэтому я пью вино.

Fig. 2. The annotation tool for laboratory evaluation

Relevance score for a query–document pair is an average over all assessors' labels. Table 4 shows exemplary stimuli and the responses of the systems along with averaged assessors' judgments. We employed Discounted Cumulative Gain (DCG) (Järvelin and Kekäläinen, 2002) as the quality metric. Table 5 shows top-1 and DCG@3 scores for each model.

Table 4. Systems' responses and their evaluation scores

Score	Stimulus	Response
2.25	ВЫ меня опять стесняетесь?	Я рожден, чтобы стесняться незнакомой компании.
	Are YOU embarrassed by me again?	I was born to be embarrassed by unfamiliar company.
2.00	А если опоздать..., что будет? :)	Если опаздываешь, не торопись. Не надо опаздывать раньше времени.
	If you're late… what will happen? :)	If you're late, do not hurry. Do not be late ahead of time.
1.75	Налить вина, совсем не грех... пить нужно лишь...)))	Грехи снимают стресс.
	It's not a sin to pour some wine… just need to drink…)))	Sins relieve stress.
1.25	Никакие редуты не помогут... когда кролик атакует?)...	Недовольный кролик =)
	No strongholds will help... when a rabbit attacks?)...	Grumpy rabbit =)

Table 5. Lab evaluation results (80 questions)

Model	top-1	DCG@3
BM25	0,76	1,48
QTR	0,85	1,58
word2vec	0,77	1,62
LETOR	0,74	1,41
Oracle	1,63	2,95

We also calculated Cohen's kappa (Carletta, 1996) as a measure of inter-annotator agreement. We used weighted variant (weights are absolute differences between labels) for pair-wise agreement. Averaged pairwise kappa statistics for four assessors in our experiments is 0.21. Example 10 illustrates QA-pair with low assessor agreement (assessor 1—😟, assessor 2—😁, assessor 3—😟, assessor 4—☺).

Example 10:

— Мысли разбежались) Как собрать—чтоб не повредить—мысли?
— «Далеко пойдешь!» — подумала мысль... и ушла.

— Thoughts have dispersed) How can they be gathered without damage?
— "You'll go far!" a thought reflected… and went away.

6. Discussion and Future Work

The results of the evaluation on the CQA platform show that the learning-to-rank approach, which was trained on the historical CQA dataset, provides the best performance. In particular, as shown in Table 3, the LETOR model is ahead of other models in terms of "likes" and "best model" measures. Moreover, it provides answers that on average have more likes than around 17% of answers provided by users of the CQA platform.

On the other hand, QTR approach has the biggest amount of "best answers" and the least amount of "likes" at the same time. The word2vec-based approach has comparable performance. We noticed that answers marked as the "best" on average have less "like" marks than other answers. This suggests that askers often disagree with the community about which answers are appropriate or funny.

The most surprising aspect of the manual evaluation is that the LETOR method shows the lowest value in both top-1 and DCG@3 metrics. There are two possible explanations for this. The first one is based on the low inter-annotator agreement. Such a low agreement confirms that the perception of humor varies greatly from person to person, and conclusive lab evaluation may require a significantly higher number of assessors. Yet another explanation of the drastic drop in the LETOR performance is that some CQA users positively evaluate answers that are not quite in the context, and thus training on the CQA data can yield a biased model. This hypothesis can be investigated in future studies by training the LETOR model using the QA pairs evaluated in laboratory settings.

The findings suggest that information retrieval approach is a promising direction in humorous response generation. It is also clear that morphological and word2vec-based features are effective for the task. Nevertheless, the results of the "oracle" model indicate that there is an abundant room for the improvement of the answer ranking. Thus, in future investigations we plan to enhance the learning-to-rank approach by incorporating features that can capture the nature of humor and context in a better way.

References

1. *Binstock A., Rex J.* (1995), Practical algorithms for programmers, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
2. *Blinov V., Bolotova V., Braslavski P.* (2016) Information retrieval approach to humorous response generation in dialog systems: a baseline, available at: http://www.dialog-21.ru/media/3462/blinov.pdf
3. *Brown P. F., Pietra V. J. D., Pietra S. A. D., Mercer R. L.* (1993), The mathematics of statistical machine translation: Parameter estimation, Computational linguistics, Vol. 19(2), pp. 263–311.
4. *Carletta J.* (1996), Assessing agreement on classification tasks: the kappa statistic, Computational linguistics, Vol. 22(2), pp. 249–254.
5. *Friedland L., Allan J.* (2008), Joke retrieval: recognizing the same joke told differently, Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, pp. 883–892.
6. *Hong B. A., Ong E.* (2009), Automatically extracting word relationships as templates for pun generation, Proceedings of the Workshop on Computational Approaches to Linguistic Creativity, pp. 24–31.
7. *Jones K. S., Walker S., & Robertson S. E.* (2000), A probabilistic model of information retrieval: development and comparative experiments: Part 2, Information processing & management, Vol. 36(6), pp. 809–840.
8. *Järvelin K., Kekäläinen J.* (2002), Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems, Vol. 20(4), pp. 422–446.
9. *Khooshabeh P., McCall C., Gandhe S., Gratch J., Blascovich J.* (2011), Does it matter if a computer jokes, CHI '11 Extended Abstracts on Human Factors in Computing Systems, Vancouver, BC, Canada, pp. 77–86.
10. *Korobov M.* (2015), Morphological analyzer and generator for Russian and Ukrainian languages, Analysis of Images, Social Networks and Texts. Communications in Computer and Information Science, Vol. 542, Springer, Cham, pp. 320–332.
11. *Kutuzov A., Kuzmenko E.* (2017), WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models, Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, Vol. 661. Springer, Cham, pp. 155–161.
12. *Lau J. H., Baldwin T.* (2016), An empirical evaluation of doc2vec with practical insights into document embedding generation, Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, pp. 78–86.
13. *Le Q. V., Mikolov T.* (2014), Distributed Representations of Sentences and Documents, Proceedings of The 31st International Conference on Machine Learning, Beijing, China, pp. 1188–1196.
14. *Mihalcea R., Strapparava C.* (2006), Technologies that make you smile: Adding humor to text-based applications, IEEE Intelligent Systems, Vol. 21(5), pp. 33–39.

15. *Mikolov T., Chen K., Corrado G., Dean J.* (2013), Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.
16. *Nijholt A.* (2002), The April Fools' Day Workshop of Computational Humour, Trento, Italy, pp. 101–111.
17. *Ratcliff J., Metzener D.* (1988), Pattern matching: The Gestalt approach, Dr. Dobb's Journal, p. 46.
18. *Ritchie G.* (2005), Computational mechanisms for pun generation, Proceedings of the 10th European Natural Language Generation Workshop, Aberdeen, Scotland, UK, pp. 125–132.
19. *Ritter A., Cherry C., Dolan W. B.* (2011), Data-driven response generation in social media, Proceedings of the conference on empirical methods in natural language processing, Edinburgh, Scotland, UK, pp. 583–593.
20. *Stock O., Strapparava C.* (2003), Getting serious about the development of computational humor, Proceedings of the 18th international joint conference on Artificial intelligence, Acapulco, Mexico, pp. 59–64.
21. *Surdeanu M., Ciaramita M., Zaragoza H.* (2011), Learning to rank answers to non-factoid questions from web collections, Computational linguistics, Vol. 37(2), pp. 351–383.
22. *Valitutti A., Toivonen H., Doucet A., Toivanen J. M.* (2013), “Let Everything Turn Well in Your Wife”: Generation of Adult Humor Using Lexical Constraints, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp. 243–248.
23. *Yang D., Lavie A., Dyer C., Hovy E.* (2015), Humor Recognition and Humor Anchor Extraction, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 2367–2376.

TEXT NORMALIZATION IN RUSSIAN TEXT-TO-SPEECH SYNTHESIS: TAXONOMY AND PROCESSING OF NON-STANDARD WORDS

Cherepanova O. D. (cherepanova.od@gmail.com)

Moscow State University, Moscow, Russia

Alongside with ordinary words, natural-language text also contains non-standard words (NSWs), such as abbreviations, acronyms, dates, phone numbers, currency amounts etc. Before phonetizing these text elements in Text-to-Speech synthesis, it is necessary to normalize them by replacing them with an appropriate ordinary word or word sequence. NSWs are increasingly diverse and most of them require specific normalization rules. In this paper, we present a taxonomy of NSWs for the Russian language developed on the basis of news texts, software and car reviews and instruction manuals. We grouped NSWs that have similar normalization rules or patterns taking into account their graphic form and their context dependence. We propose five main groups of NSWs: abbreviations (including acronyms and initialisms), text elements containing numbers, special characters, foreign words written in the Latin alphabet and mixed-type non-standard words. In this work, we describe these NSW types and address the issue of their normalization in Russian Text-to-Speech synthesis.

Key words: Text-to-Speech-synthesis, text normalization, Russian

НОРМАЛИЗАЦИЯ ТЕКСТА В СИСТЕМЕ РУССКОЯЗЫЧНОГО СИНТЕЗА «ТЕКСТ-РЕЧЬ»: КЛАССИФИКАЦИЯ И ОБРАБОТКА НЕСТАНДАРТНЫХ ТЕКСТОВЫХ ОБЪЕКТОВ

Черепанова О. Д. (cherepanova.od@gmail.com)

МГУ им. М. В. Ломоносова, Москва, Россия

Ключевые слова: синтез речи по тексту, нормализация текста, русский язык

1. Introduction

To illustrate the subject of our research let us consider the following headline: *Nokia планирует купить разработчика ПО Comptel за €347 млн* In order to correctly pronounce this sentence, a Russian Text-to-Speech synthesizer has to normalize most of these words by completing the following operations: expand the abbreviation *млн* to *миллионов*, transform the number *347* into its graphical representation *триста сорок семь*, transliterate the names *Nokia* and *Comptel* written in the Latin alphabet, classify *ПО* as an initial abbreviation and insert the word *евро* instead of the symbol €. These text units requiring additional processing rules are called ***non-standard words*** or **NSWs** ([Black etal. 1999] and others). In TTS-synthesis, NSWs should be processed at the stage of text normalization in order to get at the output "a sequence of white-space separated accentuated orthographic words" [Krivnova 1998: 5]. There is a wide range of NSWs and their number increases with every new text: they include all types of abbreviations, dates, phone numbers, addresses, special characters etc. However, many of these NSW groups have similar normalization rules.

Fig. 1 presents a basic NSW normalization algorithm based on [Sproat etal. 2001: 304] and adapted to Russian as an inflected language (Figure 1). The data used at each step (hand-written rules, dictionaries, language models etc.) depend largely on the TTS-system and can vary for different domains.

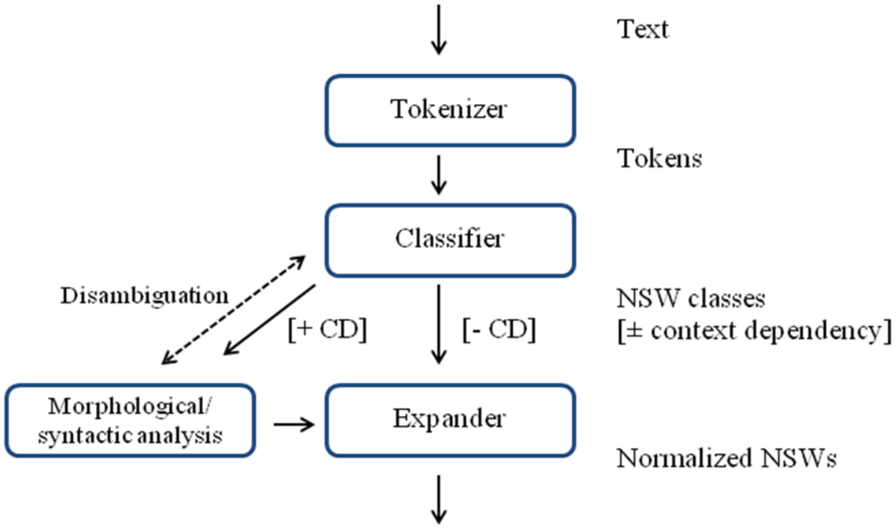


Fig. 1. Basic NSW normalization algorithm

In the following sections, we describe a NSW taxonomy designed to simplify text normalization as a stage of Russian TTS-synthesis.

2. Previous approaches

While NSW processing is described in detail for English TTS-systems (see [Black et al. 1999], [Olinsky, Black 2000], [Sproat et al. 2001]), as far as we know, there are practically no published works on this subject for the Russian language. The issue of NSW disambiguation in Russian TTS-synthesis is discussed in [Khomitsevich etal. 2013], and a detailed taxonomy of Russian abbreviations is presented in [Krivnova 1999].

A research on NSW normalization in inflected languages (on the example of Greek) was carried out by [Xydas et al. 2004] with an emphasis on NSW expansion rules.

One of the most detailed researches on NSW normalization is [Sproat etal. 2001]. The authors proposed a systematized NSW taxonomy and investigated several techniques of NSW normalization. Sproat's taxonomy provided the basis for our classification but had to be extended due to peculiarities of the Russian language. The following changes were introduced:

- a feature '± context dependency' was added to all NSW classes (since in Russian many of them require grammatical agreement within the sentence);
- a new NSW class was added for special characters (see Section 3.3);
- a new NSW class was added for words written in the Latin alphabet (Section 3.4).

3. A Taxonomy of NSWs

In our taxonomy, we tried to define NSW categories with similar normalization rules or patterns. Our taxonomy was developed on the basis of texts from news papers, car websites, software descriptions and instruction manuals.

Table 1. Taxonomy of Russian non-standard words

Class name	Context dependency	Examples
1. Abbreviations		
1.1. Shortened abbreviations	–	исполком, завлаб
1.2. Graphic abbreviations	+	филол., т.е., оз., 60 км/ч, 20 кг, пр-т
1.3. Initial letter abbreviations		
1.3.1. Initialisms	–	МГУ, СНГ
1.3.2. Acronyms	–	МГИМО, ГУМ
1.3.3. Mixed-type initial abbreviations	–	ЦСКА, ГИБДД
1.4. Mixed-type abbreviations	–	БелАЗ
2. NSWs containing numbers	+	12-ый *том*, *Иван* IV, 13:45 *часов*, 12 *тыс.* **€**
	–	*ауд.* 956, *тел.:* 8 (495) 123 45 67, *Android* 2.3
3. Special characters	+	$, **€**, **Y**, °
	–	+, −, ±, ≤, <, >, *, &, #, ~
4. Latin alphabet words	–	Windows, microSD;
5. Mixed-type NSWs	+–	№38-ФЗ, MP3-плеер

Note: 'Context dependency' means here the need for grammatical agreement of expanded NSWs (*у оз.* /озера/ *Селигер* 'near the Seliger lake' but *на оз.* /озере/ *Селигер* 'at the Seliger lake').

Taking into account our data analysis and previous taxonomies mentioned above, we propose the following taxonomy of Russian NSWs: 1) abbreviations; 2) NSWs containing numbers; 3) special characters; 4) Latin alphabet words; 5) mixed-type NSWs. The classification is based on following NSW features: graphic form of NSWs, their potential normalization rules and their context dependency. The NSW taxonomy is summarized in Table 1.

In the following sections, each NSW category is described in more detail.

3.1. Abbreviations

We use the term ‘abbreviation’ here in the broadest sense: it includes all kinds of abbreviations, acronyms and initialisms. There are several categories of abbreviations in Russian: ***1.1. shortened word combinations*** (*исполком*, *колхоз*); ***1.2. graphic abbreviations*** marked by a full stop, hyphen, slash or other graphical means (*т.е.*, *н.э.*, б/у); ***1.3. initial letter abbreviations*** formed by initial components of word combinations and pronounced in their shortened form: *МГУ* /эм-гэ-у/[1], *ГУМ* /гум/); ***1.4. mixed-type abbreviations*** (*БелАЗ*).

Most ***shortened word combinations*** (*роддом, детдом, телесеть, драмкружок, теракт, запчасть* etc.) are pronounced as ordinary words, and thus usually pose no problem in TTS-synthesis. However, abbreviations like *завлаб, местком* or *продмаг* pronounced in their shortened form might be difficult to understand; for better intelligibility they probably should be normalized to their expanded form (*заведующий лабораторией, местный комитет, продуктовый магазин*).

Graphic abbreviations have rather simple normalization rules: most of them are widely used, are generally included into dictionaries and, thus, can be verbalized in their full form. Cases of ambiguity can generally be resolved using the context: e.g., ***г.*** is expanded as /город/ ‘city’ if the previous word starts with a capital letter (*г. Москва*) or as /год/ ‘year’ after a number sequence (*2017 г.*). Graphic abbreviations can be formed in several different ways: a) by omitting the end of the word (*филол.* –/филологический/, *архит.*—/архитектурный/); b) by using initial letters of each word or syllable (*н. э., и т. д., л. с., пп., гг., вв.*); c) as abbreviations without a full stop (*км, м, кг, л, мл, т*), d) by omitting the middle of the word marked by a hyphen (*г-н, г-жа, пр-т*), e) marked by a slash (*н/Д*—/на-Дону/; *б/у*—/бывший в употреблении/). Here, a special group form abbreviations of physical units (*км/ч; об/мин; Мбит/с* etc.).

As we can see, graphic abbreviations are very diverse and, according to our data analysis, they are more frequent than other abbreviation types. As M. Rovinskaya pointed out in her undergraduate's thesis, more than the half of abbreviations marked with a full stop have only one extension variant [Rovinskaya 1998: 6]. This makes it possible to cover most graphic abbreviations by means of a dictionary and local syntactic analysis.

There are two groups of ***initial letter abbreviations***: ***1.3.1 initialisms*** (pronounced one letter at a time: *СНГ* /эс-эн-гэ/, *ГДР* /гэ-дэ-эр/); ***1.3.2 acronyms*** (pronounced as one word: *ГУМ* /гум/, *ЛЭП* /лэп/); ***1.3.3 mixed-type initial abbreviations*** (one part is pronounced as a single word, and the other—as separate letters:

[1] In the present paper we indicate proposed graphic normalization forms of NSWs by slash signs (/).

ЦСКА /цэ-эс-ка/, *ГИБДД* /ги-бэ-дэ-дэ/). In our analyzed text data, 58% of initial letter abbreviations were acronyms.

Even though we can write quite simple normalization rules for this NSW class, there is still need for a dictionary. To begin with, there can be more than one way of pronouncing the same letter: the letter 'Ф' can be pronounced both as /эф/ (*РФ*, *ФСБ*) and /фэ/ (*ФБР*, *ФРГ*). Secondly, there is a range of exceptions: e.g., *США* /сэ-шэ-а/, *ТВ* /теле/, *МЮ* /манчестер юнайтед/, and the abbreviation *МСК* which formally might be classified as an initial letter abbreviation and is pronounced as /москва/ 'Moscow' or even as a whole sentence—/по московскому времени/ 'Moscow time'.

The stress in acronyms falls usually on the last syllable, but there are some exceptions (*НАТО*, *ЮНЕСКО* etc. [Krivnova 1998: 4]) that should be included in a dictionary.

Another challenge in Russian TTS-synthesis are abbreviations written in the Latin alphabet. According to the analyzed data, English acronyms and initialisms are widely used in Russian texts and compose 29% of all initial abbreviations. A normalization method for English words and abbreviations is proposed in Section 3.4.

3.2. NSWs containing numbers

Number sequences are pronounced in different ways depending on their function—and as we can see from Table 2, there is a wide range of their functions. We defined three distinctive features for NSWs containing numbers (hereinafter 'number sequences' or 'NS'): their ***context dependency***, ***verbalization format*** and ***number class***.

We distinguish three NS categories by their verbalization format: (A) NSs pronounced as one number; (B) NSs pronounced one number at a time; and (C) NSs with special verbalization formats.

There are three number classes NSs can be expanded with: 1) cardinal numbers; 2) ordinal numbers; and 3) collective numbers.

In compliance with these features we defined 17 categories of number sequences:

Table 2. Taxonomy of NSWs containing numbers

NS category	Class[2]	Form[3]	CD[4]	Examples
2.1. cardinal numbers	C	A	+	12 домов
2.2. numbers (excluding phone numbers)	C	B	–	ауд. 956
2.3. phone numbers	C	C	–	8 (495) 123 45 67; 123-45-67
2.4. addresses	C	C	–	д.1, к.2, кв.123; д.2/3
2.5. index numbers	C	B	–	123456
2.6. time indication	C	C	+	в 13:45; к 13:45

[2] Number class: C = cardinal numbers; O = ordinal numbers; Col = collective numbers

[3] Normalization format

[4] Context dependency

NS category	Class[2]	Form[3]	CD[4]	Examples
2.7. money amounts	C	C	+	$ 1,5; 1.20 руб; 12 тыс. €
2.8. percentage	C	A	+	29,99%; 50%
2.9. series numbers	C	B	–	Android 2.3; § 4.2.3.
2.10. multiplicative constructions 'number (GEN) + adjective'	C*	A	–	11-метровый; 4-кратный; 2,0-литровый
2.11. ordinal numbers	O	A	+	12-ый том; Иван IV; 1. …; 2. …
2.12. dates	O	C	+	2.05.06; 02/05; 2 мая 2006г.
2.13. years	O	A	+	2001г.; 2010/11 гг.; 60-е
2.14. fractions	O*	A	+	1/4 финала; 2/3 опрошенных
2.15. collective numbers	Col	A	–	5-ро друзей; 2-е суток
2.16. denumerate constructions	Col*	A	–	16-ричный режим; 2-ичная нумерация
2.17. multiplicative numbers	Col*	A	+	3-ной подогрев; 4-ной сальхов

In this article, we are not going to provide a detailed review for each of the NS classes since it is a rather broad area described for the Russian language, in particular, in [Azerkovich 2013]. However, let us take a closer look at the normalization schemes of some of these NSW categories.

For better intelligibility of **money amounts**, currency units should be verbalized: *$12,25* /двенадцать долларов и двадцать пять центов/ (but: *$12,25 млн* / двенадцать целых и двадцать пять сотых миллионов долларов/). Normalization rules of currency symbols are discussed in Section 2.4.

Even though there are only a few conventional **time formats** in literary Russian, there is still room for ambiguity. For example, such NSs as *19-30*, *19.30* or *19:30* can be used not only referring to time, but also to sport scores or prices. The form *1:30* can denote day time (/час тридцать/), the length of a phone call or race time (/минута тридцать секунд/). In some cases, we can disambiguate NSs using context key words (currency names, time abbreviations etc.).

Most ordinal number NSs are pronounced as one number (group A). According to our text analysis, NSWs are expanded by ordinal numbers in date statements (*12 мая 2001 г., 2010/11 гг.*) or if there is a marked word ending (*12-ый том*, *60-е*). Ordinal numbers are also used in Roman numerals (*XIX век, Карл V, глава I, XX съезд КПСС*).

There are several **date formats**, but the standard format in Russia is 'day-month-year'. The year can be denoted both by two and four digits: *04.05.2006* and *04.05.06*. Numbers in date statements are separated by full stops (most frequent), hyphens or slashes: *04.05.2006; 04-05-2006; 04/05/2006.* In order to expand a date statement, the day number should be replaced by an ordinal number (/четвертое/), the month—by the corresponding month name in the genitive case (/мая/), and the year—by an ordinal number in the genitive case (/две тысячи шестого года/). According to our research, full date forms as listed above are used not very often. Usually, month numbers are already replaced by month names in texts (*4 мая* or *4 мая 2006 г.*). However, only the day number requires grammatical agreement with the context. Another date format that might cause difficulties is *4/5* (/четвертое мая/) since it could also denote a fraction. The normalization of slashes and other special characters is discussed in Section 3.3.

3.3. Special characters

There is a small group of context-dependent special characters requiring agreement in case and number. One of the most frequent is the percent sign **%** /процент*[5]/. It is sometimes used as an abbreviation in constructions like *10%-й раствор; 20%-я сметана*. The characters **§** /доллар*/, **№** /номер*/, **°** /градус*/, **″** /дюйм*/ and currency symbols (**€**, **$**, **£** etc.) are also context-dependent. It should be pointed out that currency symbols usually precede number sequences (€12), but should be pronounced after them: /двенадцать евро/. When used after words like *тыс., млн, млрд* etc., currency names should be normalized to their genitive form (*прибавить к 12 тыс. $ /долларов/*).

For context-independent special characters, the context can still be of paramount importance as it could be used for disambiguation. Thus, a **slash** can both denote a fraction (*2/3* — /две трети/), a division sign (*2/3 = 0.67* /два поделить на три/), a separator in physical units or date statements (*120 об/мин* — /оборотов в минуту/) and the meaning 'or' (*7 шт / 14 шт* — /семь штук **или** четырнадцать штук/). In an address, the token *2/3* can also be pronounced as /дом два дробь три/.

Semicolons are mostly used as punctuation marks, but can also denote **proportions** (*1:1000 /один **к** тысячи/; 50:50 /пятьдесят **на** пятьдесят*). **Superscript numbers** can be used as power exponents in physical units ($м^2$, $см^3$) or as footnotes. Even if there is no need in verbalizing the footnote number, a TTS-synthesizer should still recognize them in order to put in the footnote text in the right place.

A detailed taxonomy of special characters is described in Table 3.

Table 3. Taxonomy of special characters

Special character class	**Examples**	**Pronunciation**
1. Context-dependent		
§, №, %	§ 12, № 3, 20 %	/параграф* n/, /номер* n/, /n процент*/
Physical quantities (°,″)	+12 °C; −12° по Цельсию; экран 6″.	/(плюс/минус) n градус* (по Цельсию)/, /экран в n дюймов/)
Currencies	12 тыс. $, 12€, Y 150	/n тысяч долларов/, /n евро/, /n иен/
2. Context-independent		
Mathematical characters:		
+, −, ±	(+7); Google+; −0,5; ±12	/плюс/; /минус/; /плюс-минус/

[5] The symbol * marks context-dependent word endings.

Special character class	**Examples**	**Pronunciation**
*, /, :	2,5*20*5,3; 3/4; 7шт./14шт.; 1:100; 50:50; 60 км/ч	/на/; /три четвертых (три четверти)/; /или/; /к/; /в/
^, ^{2}	2^6; *м*^{2}	/x в степени n/; /метр квадратный/
=, >, <, ≤	2+2 = 4; 4>2	/равно/; /больше, чем/; /меньше, чем/; /больше или равно/
Other characters:		
&, #, @, ~, x, ©	Маркетинг&Реклама; #100; abc@web.de; 4x3; ~25%	/и/; /решетка/; /собака/; /на/; /приблизительно/
Footnotes	[1], (1), ^{2}, *, **	verbalization of the footnote text

The range of specialized characters used in texts is certainly much wider and largely depends on the topic. In the present paper, we listed only the most frequent special symbols occurring in Russian texts.

3.4. English words and word combinations in Russian texts

Nearly every Russian text contains words written in the Latin alphabet. These are mostly English names of companies, mass media, brands or software. In order to verbalize English words in a Russian TTS-system, we need to transform them into the graphic (or phonetic) system used by the synthesizer for ordinary Cyrillic words. One possible approach here is to use **orthographic transcription**. Provided that we have an IPA transcription for these English words[6], with the help of English-Russian orthographic transcription rules we can convert them into the Cyrillic alphabet with due regard to their pronunciation in English. This method is discussed in [Cherepanova 2016]; here we present only a few examples:

(1) Microsoft ['maɪkrəʊsɒft]—/ма+йкрософт/[7]

(2) British Airways ['brɪtɪʃ 'eəweɪz]—/бри+тиш э+рвэйс/.

3.4.1. Abbreviations in the Latin alphabet

There are different types of English abbreviations in Russian texts: acronyms and initialisms (*SMS, SIM, GPS, OS*), graphic abbreviations (*Ltd., Co., Inc.*), mixed-type abbreviations (*Mp3, 4G, microSD, DivX, MPEG-2, 3D, e-mail, iPhone*). Quite common are

[6] There is a large number of publicly available IPA transcription programs for the English language. However, most of them are based on dictionaries and, therefore, not all organization names can be automatically transcribed.

[7] Orthographic transcription is indicated by slashes (/), the sign '+' indicates the position of the stress.

composed constructions where English abbreviations precede Russian words: *USB-накопитель, DVD-плеер, FM-передатчик, IP-адрес*.

As it seems, graphic abbreviations should be expanded to their full form and vocalized using the same rules as for ordinary English words. Numbers used in English word constructions are always pronounced in Russian.

The verbalization of English initialisms depends on several factors. Let us compare the following abbreviations used in Russian: *3D* /три-**дэ**/ and *DVD* /**ди**-ви-**ди**/; *диск С* /диск **цэ**/ and *CD* /си-**ди**/; *MP3* /эм-**пэ**-три/ and *IP* /ай-**пи**/. Presumably, the pronunciation of such initialisms depends on their length (single letters vs. letter sequences) and usage frequency. But even the same English initialism can be pronounced in different ways: the most common pronunciation of *HTML* is /эйч-ти-эм-эль/, but there is also a rather frequent informal variant /аш-тэ-эм-эль/. The issue of intelligible and natural verbalization of English abbreviations in Russian TTS-synthesis needs further investigation (for example, an analysis of the speech of Russian news readers).

4. Conclusion

In this paper, we presented a taxonomy of non-standard words requiring special normalization rules in Russian TTS-synthesis. This taxonomy might be used in text normalization tasks and help to systematize hand-written rules of NSW processing. The presented NSW list is not intended to be exhaustive: specialized texts might contain a wide range of topic-specific abbreviations, number sequences or special characters not mentioned here. Our goal was to systematize the main NSW categories in order to provide a basis for further investigations. Moreover, our research didn't include the issue of out-of-vocabulary word processing and of spelling mistakes. Such text elements also require additional processing rules at the stage of text analysis and, in particular, were included by Sproat in one of his NSW categories. This could be an issue for further research.

References

1. *Azerkovich I. L.* (2013) Automatic identification of numbers and number groups in text normalization in Text-to-Speech synthesis [Avtomaticheskaya identifikatsiya tsifr i chislovykh grupp v protsesse normalizatsii teksta pri sinteze rechi] // Proc. XX International youth conference for students, postgraduate students and young scientists “Lomonosov-2013” [Materialy XX Mezhdunarodnoy molodezhnoy nauchnoy konferentsii studentov, aspirantov i molodykh uchenykh “Lomonosov-2013”], Moscow, MAKS Press.
2. *Black A., Chen S., Kumar Sh., Ostendorf M., Richard Ch., Sproat R., Yarowsky D.* (1999) Normalization of non-standard words, JHU99.
3. *Cherepanova O. D.* (2016) Verbalizing of English word combinations in Russian Text-to-Speech synthesis by means of orthographic transcription [Ozvuchivanie angloyazychnykh slovoupotrebleniy v sisteme russkoyazychnogo sinteza "tekst-rech" s pomoshchyu prakticheskoy transkriptsii], Online-materials of the conference “Dialogue-2016”, available at: http://www.dialog-21.ru/media/3449/cherepanovaod.pdf.

4. *Khomitsevich O. G., Rybin S. V., Anichkin I. M.* (2013) Linguistic analysis in text normalization and disambiguation in Russian text-to-speech synthesis [Ispolzovanie lingvisticheskogo analiza dlya normalizatsii teksta i snyatia omonimii v sisteme russkoy rechi]. IzvestiyaD vysshikh uchebnykh zavedeniy. Priborostroyenie, No. 2, pp. 42–46.
5. *Krivnova O. F.* (1998) Automatic Text-to-Speech synthesis (second version with a women voice) [Avtomaticheskiy sintez russkoy rechi po proizvol'nomu tekstu (vtoraya versiya s zhenskim golosom)], Proc. Int. seminar in computational linguistics and its applications "Dialogue 1998" [Trudy mezhdunarodnogo seminara po komp'yuternoy lingvistike i yeyo prilozheniyam "Dialog 1998"], Moscow.
6. *Krivnova O. F.* (1999) Processing of acronyms in automatic Text-to-Speech synthesis [Obrabotka inizialnykh abbreviatur pri avtomaticheskom sinteze rechi], Proc. Int. seminar in computational linguistics and its applications "Dialogue 1999" [Trudy mezhdunarodnogo seminara po komp'yuternoy lingvistike i yeyo prilozheniyam "Dialog 1999"], Moscow.
7. *Olinsky C., Black A. W.* (2000) Non-standard word and homograph resolution for Asian language text analysis, INTERSPEECH, ISCA, pp. 733–736.
8. *Sproat R., Black A., Chen S., Kumar Sh., Ostendorf M., Richards Ch.* (2001) Normalization of non-standard words, Computer Speech and Language, Vol. 15, pp. 287–333.
9. *Rovinskaya M. M.* (1998) Recognition of full stop functions in automatic speech synthesis [Raspoznavanie funktsii upotrebleniya tochki pri avtomaticheskom sinteze rechi], undergraduate's thesis, MSU, Moscow.
10. *Xydas G., Karberis G., Kouroupertroglou G.* (2004) Text Normalization for the Pronunciation of Non-Standard Words in an Inflected Language, Proceedings of the 3rd Hellenic Conference on Artificial Intelligence (SETN04), Samos, Greece, May 5–8.

RUSSIAN COLLOCATION EXTRACTION BASED ON WORD EMBEDDINGS[1]

Enikeeva E. V. (protoev@yandex.ru),
Mitrofanova O. A. (o.mitrofanova@spbu.ru)

Saint Petersburg State University, St. Petersburg, Russia

Collocation acquisition is a crucial task in language learning as well as in natural language processing. Semantics-oriented computational approaches to collocations are quite rare, especially on Russian language data, and require an underlying semantic formalism. In this paper we exploit a definition of collocation by I. A. Mel'čuk and colleagues (Iordanskaya, Mel'čuk 2007) and apply the theory of lexical functions to the task of collocation extraction. Distributed word vector models serve as a state-of-the-art computational basis for the tested method. For the first time experiments of such type are conducted on available Russian language data, including Russian National Corpus, SynTagRus and RusVectōrēs project resources. The resulting collocation lists are assessed manually and then evaluated by means of precision and MRR metrics. Final scores are quite promising (reaching 0.9 in precision) and described algorithm improvements yield a considerable performance growth.

Keywords: distributional semantics, compositional collocations, "Meaning ⇔ Text" theory, collocation extraction

ИСПОЛЬЗОВАНИЕ ВЕКТОРНЫХ МОДЕЛЕЙ ДЛЯ ИЗВЛЕЧЕНИЯ КОЛЛОКАЦИЙ ИЗ КОРПУСОВ РУССКОЯЗЫЧНЫХ ТЕКСТОВ

Еникеева Е. В. (protoev@yandex.ru),
Митрофанова О. А. (o.mitrofanova@spbu.ru)

Санкт-Петербургский государственный университет, Санкт-Петербург, Россия

[1] The reported study is supported by RFBR grant № 16-06-00529 "Development of a linguistic toolkit for semantic analysis of Russian text corpora by statistical techniques".

1. Introduction

Collocability is an important factor in a vast majority of natural language processing and language modelling tasks, namely, syntactic parsing, machine translation, paraphrase generation, automatic and semi-automatic dictionary acquisition, semantic role labelling, word sense disambiguation, etc. In fact, contemporary research in most fields of computational linguistics rests upon the achievements of "contextualist" framework, cf. (Khokhlova 2010, etc.).

Presumably the first definition of collocation can be found in (Palmer 1933): "A collocation is a succession of two or more words that must be learnt as an integral whole and not pieced together from its component parts". These linguistic units are usually treated as restricted co-occurrences of two (or more) syntactically bound elements (Kilgariff 2006). At the same time they should be distinguished from idioms, because target word or collocation base can co-occur with a number of other lexical units (e.g. collocations *еловая*, *сосновая*, *кедровая*, etc. *шишка* '*fir, pine, cedar*, etc. *cone*' vs. idiom *важная шишка* '*boss*'; collocations *бить тревогу*, *рекорд*, *поклоны*, etc. '*sound the alarm*, *beat the record*, *beat bows*, etc.' vs. *idioms бить баклуши* '*twiddle*').

In our study much attention is given to the treatment of collocations in "Meaning ⇔ Text" theory (Iordanskaya, Mel'čuk 2007, Mel'čuk 1998) reflected in Explanatory Combinatorial Dictionary of Contemporary English (Mel'čuk, Zholkovsky 1984) and in SynTagRus Treebank (http://www.ruscorpora.ru/search-syntax.html). The theory allows to describe collocation structure in terms of lexical functions (LFs) that associate one lexical unit (argument, base) with another (value, collocate) which is selected by the rules of a language to express a meaning of given LF (cf. Section 3.1. below). Therefore, it is obvious that collocations are language-specific, for example, a meaning of '*do*, *perform*' for a base '*lecture*' is in English expressed by a lexeme '(*to*) *hold*' while in Russian the same meaning is conveyed by '*читать*' ('*chitat"*, '(*to*) *read*'). Significance of "Meaning ⇔ Text" approach rests on the idea that collocations are expected to reveal both syntagmatic unity and lexical correlation of its parts.

In recent years we have witnessed rapid expansion of various collocation extraction techniques, which are based on co-occurrence statistics. Automatic tools for collocation extraction usually produce a list of ranked bigrams or n-grams. The ranking (reflecting the so called 'collocation strength') is obtained in most cases by means of a statistical association measure such as t-score or PMI. Morphosyntactic annotation of processed corpora allows to bring into action such linguistic data as lexical-syntactic patterns and/or valency frames defining boundaries of syntactic groups for collocations and possibly their inner argument structure (e.g., Word Sketch Engine, https://www.sketchengine.co.uk/; RNC Sketches, http://ling.go.mail.ru/synt/). However, even the most sophisticated techniques of collocation extraction fail to take into account lexical and semantic peculiarities of collocations.

The purpose of our study is to prove the possibility of LF-oriented automatic collocation extraction for Russian. Our aim is to extract sets of collocations for target LFs from large corpora by means of machine learning. In this paper we try to combine a formal theory of collocations in "Meaning ⇔ Text" theory with distributed word representations (Mikolov et al. 2013a). Distributed word vectors can be used to extract linguistic regularities from a large corpus in an automated way. We are the first to perform experiments for Russian corpora in the given settings. The expected output is fine-grained classification of collocations according to their lexical meaning and syntactic structure which is important both in language learning and NLP applications (Apresjan et al. 2002; Kolesnikova, Gelbukh 2012, etc.).

The paper is structured as follows: first of all, an outline of the research in the field is presented. Then, we briefly describe a theoretical background of our study and present a computational model. In the following sections we present experimental settings and evaluation framework, and conclude with the results and its discussion.

2. Related work

Although publications on statistical collocation extraction seem to be overwhelming, there is much to be done in this field. In this section we mention several remarkable studies on Russian data. As observed in the introduction, most of approaches implemented and applied to Russian text corpora do not take into account the semantic structure of collocations, describing more or less free word combinations alongside with idioms, for example, in (Yagunova, Pivovarova 2010) collocations *сердечный приступ* '*heart attack*' and *круглый стол* '*round table*' take neighboring positions in the list.

The most popular statistical measures used to compute word association within collocations include Mutual Information (MI), Dice coefficient, Log-Likelihood and t-score (Khokhlova 2010). Multiword expressions scoring by means of learning-to-rank methods involved in information retrieval is discussed in (Tutubalina, Braslavski 2016). The approach is based on machine learning techniques, it makes use of bigrams from dictionaries as training data and, as authors say, treats collocations, idioms, set phrases in a uniform way. In (Kormacheva et al. 2014) six metrics (frequency, refined frequency ratio, weighted frequency ratio, MI, Dice score and t-score) were tested on Russian prepositions and the best performance was shown by refined frequency ratio score. Previously mentioned (Yagunova, Pivovarova 2010) compared MI and t-score and prove that the former is more suitable for extracting collocations reflecting domain-specific terms. The latter (t-score) gives preference to phrases that may be called auxiliary (two-word parentheses, discourse phrases).

The recent results of Collocations, Colligations and Corpora Project (CoCoCo, (Kopotev et al. 2015)) are presented in (Kormacheva et al. 2016). Collocations and colligations are classified according to the association between phrase constituents: some of them are marked as idioms and others are subject to semantic generalization: e.g., *sleight of* [*hand/mouth/mind*]. The procedure is fully automated and based on multiple grammatical and lexical features.

A study concerning association strength measurement in syntactic constructions and testing methodology is described in (Bukia et al. 2015). The authors study adjective-noun collocations, and their algorithm predicts association even for the combinations absent from corpus. Verb-noun collocation extraction from Russian texts is studied in (Akinina et al. 2013). The approach is PMI-based and takes into account syntactic information without further semantic classification.

Association strength measurement is closely related to identification of abnormal lexical compositions (Vecchi et al. 2011) and automatic lexical error detection

(Kochmar, Briscoe 2013). The latter work presents a number of semantic anomaly measures in a vector space. A distributional approach applied to Russian error correction in collocations can be found in (Panicheva, Mitrofanova 2016).

Compositional distributional semantics provides successful solution of the collocation extraction task. As far as evidence for Russian is concerned, several vector models were evaluated during RUSSE workshop (Panchenko et al. 2015). Nowadays attention of researchers working with distributed word vector representations for Russian is focused on RusVectōrēs (Kutuzov, Kuzmenko 2017), AdaGram (Bartunov et al. 2015) and RDT (Panchenko et al. 2016) models. However, semantic relatedness evaluation only involves paradigmatic relations between lexical units. In (Bukia et al. 2016) two distributional approaches to selectional preference modelling are compared. The first one implies semantic similarity calculation based on cosine distance; the second one relies on Mikolov's (Mikolov 2013b) assumption about linguistic regularities captured by distributed word vector models. The metric similar to the latter is used as a baseline in (Rodríguez-Fernández et al. 2016): collocates are evaluated against a difference between example headword and collocate added to test headword. The main method proposed in the same paper is based on linear transformation between headword and collocate space. The approach is tested on manually classified samples drawn from Macmillan Collocations Dictionary (Rundell 2010).

Our study follows the experience of our colleagues and takes into account peculiarities of Russian data and resources.

3. Theoretical model

3.1. Collocations in "Meaning ⇔ Text" theory

"Meaning ⇔ Text" theory provides an exhaustive analysis of phraseological expressions, taking into account various types of interaction between lexical and semantic components constituting the meaning of a word group as well as well as syntactic relations established between co-occurring words. Collocations are considered as a subclass of non-free utterances (or phrasemes). A formal definition of collocations (or semi-phrasemes) runs as follows: "A collocation **AB** of language **L** is a semantic phraseme of **L** such that its signified 'X' is constructed out of the signified of the one of its two constituent lexemes—say, of **A**—and a signified '*C*' ['*X*' = '*A* ⊕ *C*'] such that the lexeme **B** expresses 'C' contingent on **A**" [Mel'čuk 1998: 30]. A collocation includes a base constituting a freely chosen semantic nucleus of a word group and a collocate being a restricted component which determines the meaning of the whole as a function of the base. Opposite to idioms which reveal non-compositional nature, collocations are treated as compositional phrasemes conforming to lexical constraints imposed on collocates (*сильный акцент* '*heavy accent*', *високосный год* '*leap year*', *спать глубоким сном* '*be soundly asleep*' и т. д.)

In case of restricted lexical co-occurrence the relations between a base and a collocate reproduced in semantically and syntactically similar expressions are represented by lexical functions (LFs) which are formally defined as follows: $f(A) = B$, for example, MAGN(*болезнь* '*disease*') = *тяжелая* ('*serious*').

Some of the most frequent syntagmatic LFs are:

- MAGN means ‘very’, ‘to a (very) high degree’, ‘intense(ly)’: MAGN(смеяться ‘laugh‘) = от души ’heartily';
- OPER1 introduces a support verb meaning ‘do’, ‘perform’: OPER1(поддержка ‘support’) = оказывать ‘(to) lend’;
- FUNC0 means that an event described by a headword takes place: FUNC0(снег ‘snow’) = идёт ‘falls’, etc.

In fact, LFs describe not only syntagmatic relations but also paradigmatic (SYN(*врач ‘doctor, physician*’) = *доктор* (‘*doctor, physician*’), ANTI(быстрый ‘fast’) = медленный ‘slow’, CONV(покупать ‘buy’) = *продавать* ‘*sell*’, etc.), and derivational ones (S0(*гордый* ‘*proud*’) = *гордость* (‘*pride*’), A1(*голод* ‘*hunger*’) = *голодный* ‘*hungry*’, CAUS(*понимать* ‘*understand*’) = *объяснять* ‘*explain*’, etc.).

LFs of different types can be combined in complex functions to express one meaning:

> INCEPOPER1 = INCEP (= ‘*to start*’) × OPER1: INCEPOPER1(*привычка* ‘*habit*’) = *приобретать* ‘*acquire*’.

At present the inventory of LFs comprises 116 varieties of standard and non-standard LFs (Apresjan et al. 2007). Russian collocations revealing LF relations are thoroughly described in the Explanatory Combinatorial Dictionary of Modern Russian (Zholkovsky, Melchuk 1984) and annotated in Russian National Corpus (RNC) subcorpus SynTagRus (Frolova, Podlesskaya 2011).

3.2. Predicting LF values by means of vector model

Mikolov and colleagues (Mikolov et al. 2013b) prove that regular linguistic relations between two word spaces may be described as a linear transformation on them. In case of collocations the relation to be modelled is perfectly formalized as a lexical function in “Meaning ⇔ Text” theory.

Our task is to predict values of a particular LF for an argument in question given training instances of this LF. Following (Rodríguez-Fernández et al. 2016), we define an argument space *A* and collocate space *C* produced by word2vec toolkit. Let *T* be a set of collocations t_i comprising argument-value pairs (a_{t_i}, c_{t_i}), that represent a given lexical function *L*. Argument matrix $A_T = [a_{t_1}, \dots, a_{t_n}]$ and collocate matrix $C_T = [c_{t_1}, \dots, c_{t_n}]$ are made up of corresponding word vectors. Then, given examples of a particular LF (e.g., MAGN: *тяжёлая болезнь* ‘*hard illness*’, *сильный акцент* ‘*heavy accent*’, etc.), we should find a transformation which converts an argument vector to a value vector of this LF, for instance, predicts a collocate *бурный* ‘*wild*’ (MAGN value) for an argument *аплодисменты* '*applause*’.

A linear transformation matrix $\Psi \in \mathbb{R}^{B\times C}$ learnt from training set *T* satisfies the following: $A_T\Psi_\mathrm{T} = C_T$.

Therefore, Ψ can be approximated using singular value decomposition to minimize the sum:

$$\sum_{i=1}^{|T|}\left\|\Psi_T a_{t_i} - c_{t_i}\right\|^2.$$

Thus, we obtain a transformation matrix for a given LF. Applying it (multiplying it by argument vector representation) we obtain a ranked list of potential collocates for a given headword and lexical function. Following (Rodríguez-Fernández et al. 2016) we then use part-of-speech collocation patterns and NPMI filters. NPMI stands for normalized pointwise mutual information and is calculated as follows:

$$NPMI = \frac{PMI(a,c)}{-\log p(c)}.$$

4. Experiments

4.1. Test data

The resources containing LF markup for Russian language are quite limited. In our experiments we use SynTagRus Treebank (http://www.ruscorpora.ru/instruction-syntax.html)[2] and Verbal collocations of Russian abstract nouns dictionary (http://dict.ruslang.ru/abstr_noun.php). SynTagRus is a subset of Russian National Corpus (http://www.ruscorpora.ru) where each sentence is assigned a parse tree as well as a list of LFs in "Meaning ⟺ Text" notation. Verbal collocations dictionary uses its own markup scheme based on LF inventory. Collocations are classified in terms of 'regular abstract meanings', such as necessity, existence, action, with additional labels such as phase (start, finish) or semantic class (cognition, perception etc.)

The authors of (Rodríguez-Fernández et al. 2016) have proved their assumption that headword and collocate embeddings should be trained on different corpora. In their work headword vectors are obtained from a small corpus containing primarily literal usage (Wikipedia), while collocate vectors are trained on a large corpus full of various figurative meanings. Our experiments are aimed at testing this hypothesis once again on available Russian language data. Thus, we use precomputed word embeddings by RusVectōrēs project (http://ling.go.mail.ru/ru/, version 3) trained on Russian National Corpus and Web corpus. NPMI scores are precomputed on Russian fiction corpus (collected from M. Moshkov's library, URL: http://lib.ru).

Table 1. Lexical functions and its frequency

LF	argument	value	Syntagrus frequency	gloss in (Rodríguez-Fernández et al. 2016)	rank in (Rodríguez-Fernández et al. 2016)
OPER1	цель 'aim'	иметь 'have'	818	'perform'	2
MAGN	каблук 'heel'	высокий 'high'	799	'intense'	1

[2] We are deeply grateful to SynTagRus team, especially to Leonid L. Iomdin and colleagues form IPPI RAS, for providing access to the data on LF.

LF	argument	value	Syntagrus frequency	gloss in (Rodríguez-Fernández et al. 2016)	rank in (Rodríguez-Fernández et al. 2016)
CAUSFUNC0	соревнование ‘competition’	проводить ‘hold’	256	—	—
FUNC0	открытие ‘opening’	состояться ‘be held’	226	—	—
INCEPOPER1	работа ‘work’	приступать ‘start’	210	‘begin to perform’	4
OPER2	правка ‘correction’	подвергаться ‘undergo’	140	—	—
REAL1-M	ракета ‘rocket’	запускать ‘launch’	109	—	—
REAL1	средства ‘means’	расходовать ‘spend’	97	—	—
INCEPFUNC0	речь ‘conversation’	заходить ‘turn to’	94	—	—

4.2. Test setup and evaluation

First of all, we conducted experiments on 9 most frequent LFs[3] from SynTagRus (table 1). In the table we present also semantic glosses from (Rodríguez-Fernández et al. 2016) and its frequency ranks for comparison. It is quite surprising that LFs’ frequency distribution in Russian corpus differs from Macmillan Collocations Dictionary. An initial collocation set was extracted from the treebank and 10 headwords of these collocations were randomly chosen as a test set. The remaining part comprises a training set. For each LF top-10 ranked collocates were assessed manually. The performance was then evaluated using precision and mean reciprocal rank (MRR) on this list of 10 collocates:

$$precision = \frac{tp}{tp + fp}.$$

where *tp* is a number of correct collocates among the retrieved list, *fp* is a number of false collocates in the list;

$$precision' = \frac{ep}{e},$$

where *ep* is a number of expected collocates (found in SynTagRus) among the top-10 retrieved ones and *e* is a number of collocates found in SynTagRus;

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{ran},$$

[3] LOC lexical function was excluded from top-10 as its value is expressed by preposition

where Q is the top-10 list and $rank_i$ is a rank of the first correct collocate (according to experts' annotation).

Filtering was conducted using Universal Dependencies part-of-speech tags assigned to lexemes in RusVectōrēs models. As seen from table 1, all test lexical functions have verbal values, so POS tags filtering in our case means simply eliminating collocates with other POS tags. NPMI threshold was chosen experimentally on some headwords different from testset.

Following (Rodríguez-Fernández et al. 2016) we present several models:

- M1—a baseline vector model from (Bukia et al. 2016) which is virtually the same as a baseline in (Rodríguez-Fernández et al. 2016). For each candidate collocate we compute its cosine similarity to $vec(a_i) - vec(c_i) + vec(a_j)$, where (a_j, c_i) is an example collocation for a given LF and a_j is a test headword;
- M2—the same baseline filtered by POS tags and NPMI scores;
- M3—the model described above using the same vector spaces for headwords and collocates trained on RNC;
- M4—model M3 filtered by POS tags and NPMI scores;
- M5—model M3, but collocate vectors are obtained from Russian Wikipedia corpus;
- M6—model M5 filtered by POS tags and NPMI scores.

Table 2. Precision scores

LF	***M1***	***M2***	***M3***	***M4***	***M5***	***M6***
OPER1	0.11	0.31	0.10	0.14	0.37	**0.63**
MAGN	0.23	0.28	0.24	0.28	0.63	**0.84**
CAUSFUNC0	0.10	0.33	0.22	0.23	0.54	**0.64**
FUNC0	0.21	0.40	0.29	0.33	**0.42**	**0.42**
INCEPOPER1	0.10	0.38	**0.64**	**0.64**	0.15	0.15
OPER2	0.17	0.28	0.12	0.11	0.29	**0.39**
REAL1-M	0.20	**0.66**	0.24	0.26	0.40	0.52
REAL1	0.15	0.37	0.32	0.33	**0.66**	**0.66**
INCEPFUNC0	0.13	0.28	0.24	0.23	0.35	**0.43**

Table 3. Expected precision scores

LF	***M1***	***M2***	***M3***	***M4***	***M5***	***M6***
OPER1	0.50	0.50	0.48	**0.65**	0.57	**0.65**
MAGN	0.68	0.70	0.70	0.70	**0.83**	0.78
CAUSFUNC0	0.33	0.45	0.87	**0.9**	0.80	0.80
FUNC0	0.70	0.80	**0.90**	0.81	0.58	0.58
INCEPOPER1	0.40	**0.55**	0.50	0.50	0.50	0.50
OPER2	0.55	0.60	0.53	0.52	**0.75**	0.70
REAL1-M	0.55	0.70	**0.87**	**0.87**	0.70	0.75
REAL1	0.40	0.45	**0.73**	0.70	0.60	0.60
INCEPFUNC0	0.50	0.70	0.72	0.67	**0.82**	0.77

Table 4. MRR scores

LF	*M1*	*M2*	*M3*	*M4*	*M5*	*M6*
OPER1	0.22	0.55	0.11	0.48	0.34	**0.73**
MAGN	0.30	0.68	0.30	0.68	0.50	**0.90**
CAUSFUNC0	0.11	0.43	0.37	0.76	0.41	**0.82**
FUNC0	0.30	0.82	0.47	**0.89**	0.36	0.64
INCEPOPER1	0.11	0.60	0.01	**0.64**	0.15	0.48
OPER2	0.19	0.64	0.23	0.48	0.37	**0.66**
REAL1-M	0.08	0.77	0.36	0.70	0.34	**0.86**
REAL1	0.23	0.50	0.37	**0.70**	0.30	0.59
INCEPFUNC0	0.10	0.64	0.37	**0.73**	0.42	0.72

4.3. Discussion

The results are presented in tables 2–4. As expected, the presented models outperform the baseline except for several cases. In general, a considerable improvement in precision and MRR scores is achieved by filtering. On the other hand, as far as precision on expected collocations is concerned, NPMI filters discard some relevant examples, so that the scores without filtering are higher. As regards ranking (assessed by MRR metric), we do not observe a steady improvement when using a different corpus to model collocate vector space on several test LFs: FUNC0, INCEPOPER1, REAL1, INCEPFUNC0. We suppose, that collocates corresponding to these LFs' values are quite rare in general domain corpus. On the contrary, as these meanings are quite specific (not abstract), they are better represented in standard register corpus.

It should be mentioned, that there is a number of headwords where an expected LF value is absent from a retrieved top-10 list. However, in the majority of such lists there is at least one correct collocate.

The examples of ranked collocates are presented in table 5. Correct collocates corresponding to target LF values are underlined. Correct collocates coinciding with the expected LF values are given in bold type. More frequent LF collocates (MAGN, FUNC0) generally seem to be retrieved with higher precision because of wider collocability of its arguments and their high frequency. On the other hand, more specific LFs (REAL1, INCEPOPER1) are also processed correctly because such combinations are quite specific and usually both headword and collocate occur in quite specific contexts.

Table 5. Retrieved collocates examples. Correct LF values are underlined

headword	LF (Mel'čuk, Zholkovsky 1984)	retrieved collocates
довод	MAGN(*довод*)= *убедительный*	*решительный*, ***убедительный***, *основательный*, *веский*, *главный*, *бесспорный*, *достаточный*…

headword	LF (Mel’čuk, Zholkovsky 1984)	retrieved collocates
домино	OPER1(*домино*) = *играть*	***играть***, *поиграть*, *стучать*, *резаться*, *сыграть*, *игра*, *бильярд*, *футбол*…
арест	OPER2(*арест*) = *сидеть*	*подвергаться*, *находиться*, *подвергать*, *брать*, *миновать*, *попадать*…
азарт	INCEPOPER1(*азарт*) = *входить*	*приходить*, *игра*, *увлекаться*…
дорога	FUNC0(*дорога*) = *проходить*	*идти*, *пойти*, *тянуться*, *лежать*, *плестись*, *тащиться* …
день	INCEPFUNC0(*день*) = *наставать*	*наступать*, ***наставать***, *начинаться*, *приходить*, *прийти*, *намечаться*, *длиться*, *заканчиваться*…
встреча	CAUSFUNC0(*встреча*) = *назначать*, *проводить*, *устраивать*	***назначать***, *намечать*, ***проводить***, *уславливаться*, *потребовать*, *приглашать*…
газета	REAL1(*газета*) = *читать*	***читать***, *прочитывать*, *перечитывать*, *почитывать*, *читывать*, *листать*, *писать*, *пересказывать*…
долг	REAL1-M(*долг*) = *выполнять*, *исполнять*, *отдавать*	***исполнять***, *погашать*, ***исполнить***, *уплачивать*, *погасить*, ***отдавать***, *заплатить*, *повиноваться*, *обязывать*…

Alongside with correct LF values the output also includes several erroneous cases. First of all, some of the potential candidates do not represent a typical value of a given LF, for example, *главный довод* ‘*main reason*’ may be treated as a realization of MAGN, though the sense of ‘intense’ is not the main one in the given phrase. Secondly, virtually all of the retrieved words may be interpreted as values of a lexical function (not necessarily the target LF). Consider the case of the headword ‘*ошибка*’. The lexemes *неточность, просчёт, промах, погрешность, дефект, описка* represent synonyms (SYN(*ошибка*)); *допускать* is a possible value of OPER1(*ошибка*). Other cases are *квартира* = COHYP(*дом*) given REAL1(*дом*) = *жить*; *долгий* = MAGN(*разбор*); *эскадра* = MULT(*корабль*), etc.

Thus, negative examples, although they go beyond the scope of our study, yield consistent explanation in terms of LF theory. Our data provide evidence on the possibility of retrieving “bundles” of lexical functions for a given word, e.g. FUNC0(*речь*) = *идти*: OPER1(*речь*) = *произносить*, HYPO(*речь*) = *тирада*, *скороговорка*, S1(*речь*) = *оратор*, S-LOC(*речь*) = *митинг*, *банкет*, VER(*речь*) = *застольная*, etc. Thorough description of LF “bundles” obtained for a given headword allows to bridge a gap from the pure collocation analysis to the complex study of lexical-syntactic constructions.

5. Conclusion

Our study shows that enrichment of traditional statistical techniques of collocation extraction by means of vector space models and lexical-syntactic information (in our case, LF data) gives new insights into the problem of how word meanings interact in contexts. In most cases contemporary corpus-based data available for Russian ignore lexical structure of collocations and provide statistical information based on association measures and/or morphosyntactic patterns. On the one hand, by now profound semantic analysis has been performed for more complex linguistic units registered in Russian corpora, namely, constructions (cf. Lexicograph (URL: http://lexicograph.ruslang.ru/) and FrameBank (URL: http://framebank.ru/) projects). On the other hand, description of fine-grained lexical-semantic relations of LF type has been carried out within the lexicographic framework, given a limited list of headwords and narrow set of their collocations which maintain certain LFs (Mel'čuk, Zholkovsky 1984).

Our research is the first to provide reliable evidence on the possibilities of automated retrieval and classification of collocations exhibiting LFs in large corpora of Russian, thus bridging the gap between traditional dictionaries and corpus-based semantic representations. We have successfully applied a state-of-the-art approach (distributed word vector representations) to extracting potential collocates for given headwords and target lexical functions. The method requires only a dictionary of tagged collocations (the SynTagRus corpus with LF markup, in our case) and corpora for distributed representation learning. Since these corpora are quite large, the number of retrieved collocates exceeds the number of collocates listed in the dictionary. The approach discussed and tested in our paper promises a vast field for future research.

We are going to improve experimental settings. In reported research we used word embeddings data from word2vec models in RusVectōrēs project, now we've got the possibility to use AdaGram (URL: https://github.com/sbos/AdaGram.jl) and/or RDT (URL: https://nlpub.ru/Russian_Distributional_Thesaurus) models. As experiments were carried out for 9 most frequent LFs and 10 randomly chosen headwords, it is also reasonable to expand the input data and to obtain a large list of LF-specific collocations.

We consider several applications of results achieved in course of experiments. A list of LF-specific collocations may be used in a set of computational semantics tasks requiring co-occurrence data: lexicon expansion for machine translation tasks (Protopopova et al. 2015), fact extraction and opinion mining (Protopopova et al. 2016), psycholinguistic profiling (Panicheva et al. 2016), automatic topic labelling, bigram-based topic modelling (Mirzagitova, Mitrofanova 2016), etc.

References

1. *Akinina, Y., Kuznetsov I., Toldova, S.* (2013) The impact of syntactic structure on verb-noun collocation extraction. In: Computational Linguistics and Intellectual Technologies, Proceedings of the Annual International Conference "Dialogue". Pp. 2–17.
2. *Apresjan, Ju. D., Boguslavsky, I. M., Iomdin, L. L., Tsinman, L. L.* (2002) Lexical Functions in NLP: Possible Uses. In: Computational Linguistics for theNew Millenium: Divergence or Synergy? In: Proceedings of the International Symposium held at the Ruprecht-Karls-Universität Heidelberg, 21–22 July 2000. Manfred Klenner / Henriëtte Visser (eds.) Frankfurt am Main. Pp. 55–72.
3. *Apresjan, Ju. D., Djachenko, P. V., Lazursky A. V., Tsinman L. L.* (2007) On the Digital Textbook on Russian Lexica [O kompjuternom uchebnike russkogo jazyka]. In: The Russian Languiage in a Scientific Light [Russkij jazyk v nauchnom osveschenii], vol. 2(14). Pp. 48–112.

4. *Bartunov, S., Kondrashkin, D., Osokin, A., Vetrov, D.* (2015) Breaking Sticks and Ambiguities with Adaptive Skip-gram. ArXiv preprint.
5. *Bukia, G. T., Protopopova, E. V., Panicheva, P. V., Mitrofanova, O. A.* (2016) Estimating Syntagmatic Association Strength Using Distributional Word Representations. In: Computational Linguistics and Intellectual Technologies, Proceedings of the Annual International Conference "Dialogue", 2016. Vol. 15. pp. 112–122.
6. *Bukia, G., Protopopova, E., Mitrofanova, O.* (2015) A corpus-driven estimation of association strength in lexical constructions. In: Sergey Balandin, T. T., Trifonova, U.(eds.) Proceedings of the AINL-ISMW FRUCT. pp. 147–152. FRUCT Oy, Finland, http://fruct.org/publications/ainl-abstract/files/Buk.pdf
7. *Frolova, T. I., Podlesskaia, O. Ju.* (2011) Tagging Lexical Functions in Russian Texts of SynTagRus Tagging lexical functions in Russian texts of SynTagRus. In: Computational Linguistics and Intellectual Technologies, Proceedings of the Annual International Conference "Dialogue". 2011. Vol. 10(17). Pp. 219–230.
8. *Khokhlova, M. V.* (2010) The Study of lexical-syntactic co-occurrence in Russian by statistical methods (based in text corpora) [Issledoanie leksiko-sintaksicheskoj sochetaemosti v russkom jazyke s pomoschju statisticheskih motodov]. PhD Thesis. Saint-Petersburg, 2010.
9. *Kilgarriff, A.* (2006) Collocationality (and how to measure it). In: Proceedings of the Euralex International Congress.
10. *Kochmar, E., Briscoe, T.* (2013) Capturing anomalies in the choice of content words in compositional distributional semantic space. In: RANLP. Pp. 365–372.
11. *Kolesnikova, O., Gelbukh A.* (2012) Semantic relations between collocations—A Spanish case study. Vol. 45, No. 78. Pp. 44–59.
12. *Kopotev, M. Escoter L., Kormacheva, D., Pierce, M., Pivovarova, L., Yangarber, R.* (2015) CoCoCo: Online Extraction of Russian Multiword Expressions. In: Proceedings of the 5th Workshop on Balto-Slavic Natural Language Processing BSNLP 2015. Hissar, Bulgaria. Pp. 43–45.
13. *Kormacheva, D., Pivovarova, L. & Kopotev, M.* (2014) Automatic Collocation Extraction and Classification of Automatically Obtained Bigrams. In: Proceedings: Workshop on Computational, Cognitive, and Linguistic Approaches to the Analysis of Complex Words and Collocations (CCLCC 2014). Pp. 27–33.
14. *Kormacheva, D., Pivovarova, L. & Kopotev, M.* (2016) Constructional generalization over Russian collocations. In: Mémoires de la Société néophilologique de Helsinki.
15. *Kutuzov, A., Kuzmenko, E.* (2017) WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. In: Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science. Vol. 661. Springer.
16. *Mel'čuk, I., Zholkovsky A.*(1984) Explanatory Combinatorial Dictionary of Modern Russian [Tolkovo-kombinatornyj slovar russkogo jazyka]. Vienna, 1984.
17. *Mel'čuk, I. A.* (1998). Collocations and Lexical Functions. In: Anthony P. Cowie (ed.) Phraseology. Theory, analysis, and applications. Pp. 23–53. Oxford: Clarendon.
18. *Mikolov T., Yih, W.-t., and Zweig G.* (2013) Linguistic Regularities in Continuous Space Word Representations. In: Proceedings of NAACL HLT.

19. *Mikolov, T., Chen, K., Corrado, G., and Dean, J.* (2013) Efficient Estimation of Word Representations in Vector Space. In: Proceedings of Workshop at ICLR.
20. *Mirzagitova, A., Mitrofanova, O.* (2016) Automatic assignment of labels in Topic Modelling for Russian Corpora. In: Proceedings of 7th Tutorial and Research Workshop on Experimental Linguistics, ExLing 2016 / ed. A. Botinis.—Saint Petersburg: International Speech Communication Association.
21. *Palmer, H. E.* (1933) Second Interim Report on English Collocations, Tokyo: Institute for Research in English Teaching.
22. *Panchenko, A., Loukachevitch, N., Ustalov, D., Paperno, D., Meyer, C., Konstantinova, N.* (2015) RUSSE: The first workshop on Russian semantic similarity. In: Computational Linguistics and Intellectual Technologies, Proceedings of the Annual International Conference "Dialogue". Pp. 89–105.
23. *Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N. and Biemann, C.* (2016) Human and Machine Judgements about Russian Semantic Relatedness. In: Proceedings of the 5th Conference on Analysis of Images, Social Networks, and Texts (AIST'2016). Communications in Computer and Information Science (CCIS). Springer-Verlag Berlin Heidelberg.
24. *Panicheva, P., Bogolyubova, O., Ledovaya, Y.* (2016) Revealing Interpetable Content Correlates of the Dark Triad Personality Traits. In: RUSSIR-2016. Russia, Saratov. Springer.
25. *Panicheva, P., Mitrofanova, O.* (2016) Developing a Toolkit for Distributional Analysis of Abnormal Collocations in Russia. In: Proceedings of the 13th Conference on Natural Language Processing (KONVENS), 2016. pp. 203–208.
26. *Protopopova, E, Bukia, G., Mitrofanova, O.* (2016) Sentiment analysis of reviews based on automatically developed lexicon. In: Proceedings of the 45th International Philological Conference. St. Petersburg State University, March 2016.
27. *Protopopova, E., Antonova, A., Misyurev, A.* (2015) Acquiring relevant context examples for A translation dictionary. In: Computational Linguistics and Intellectual Technologies, Proceedings of the Annual International Conference "Dialogue".
28. *Rodríguez-Fernández, S., Anke, L., Carlini, R., Wanner, L.* (2016) Semantics-driven recognition of collocations using word embeddings'. In: Proceedings of the 2016 Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany.
29. *Rundell, M.* (2010) Macmillan Collocations Dictionary, Macmillan.
30. *Tutubalina, E., Braslavski, P.* (2016) Multiple Features for Multiword Extraction: a Learning-to-Rank Approach, Computational Linguistics and Intellectual Technologies. In: Proceedings of the Annual International Conference "Dialogue".
31. *Vecchi, E. M., Baroni, M., Zamparelli, R.* (2011) (linear) maps of the impossible: capturing semantic anomalies in distributional space. In: Proceedings of the Workshop on Distributional Semantics and Compositionality.
32. *Yagunova, E. V., Pivovarova, L. M.* (2010) The nature of collocations in the Russian language. The experience of automatic extraction and classification of the material of news texts. In: Automatic Documentation and Mathematical Linguistics, vol. 44, issue 3, Springer. Pp. 164–175.

COMPARATIVE ANALYSIS OF ANGLICISM DISTRIBUTION IN RUSSIAN SOCIAL NETWORK TEXTS

Fenogenova A. S. (alenush93@gmail.com),
Karpov I. A. (karpovilia@gmail.com),
Kazorin V. I. (zhelyazik@mail.ru),
Lebedev I. V. (innlebedev@gmail.com)

National Research University Higher School of Economics,
Research and Development Institute KVANT, Moscow, Russia

Due to the process of globalization, the number of English borrowings in different languages is constantly growing. In natural language processing (NLP) systems, such as spell-check, POS tags, etc. the analysis of loan words is not a trivial task and should be resolved separately. This article continues our previous work on the corpus-driven Anglicism detection by proposing an improved method to the search of loan words by means of contemporary machine translation methods. It then describes distribution of the borrowed lexicon in different online social networks (OSN) and blog platforms showing that the Anglicism search task strongly depends on corpus formation method. Our approach does not contain any pre-prepared, manually acquired data and gives a significant automation in Anglicism dictionary generation. We present an effective dictionary collection method that gives the same coverage compared to random user selection strategy on a 20 times smaller corpus. Our comparative study on LiveJournal, VKontakte, Habrahabr and Twitter shows that different social, gender, even age groups have the same proportion of Anglicisms in speech.

Key words: Anglicisms, distributive semantics, social media texts, semantics, vector representation

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАСПРЕДЕЛЕНИЯ АНГЛИЦИЗМОВ В РУССКИХ ТЕКСТАХ СОЦИАЛЬНЫХ МЕДИА

Феногенова А. С. (alenush93@gmail.com),
Карпов И. А. (karpovilia@gmail.com),
Казорин В. И. (zhelyazik@mail.ru),
Лебедев И. В. (innlebedev@gmail.com)

Национальный исследовательский университет
Высшая школа экономики, Научно-исследовательский
институт «Квант», Москва, Россия

В связи с процессом глобализации, наблюдается рост количества английских заимствований во многих языках мира. Поиск подобных заимствований представляет интерес как для теоретических исследований в области языковых контактов и межъязыкового взаимодействия, так и в прикладных задачах, например при разработке средств морфологического анализа, исправления опечаток и машинного перевода.

Данная работа продолжает выполненное авторами ранее исследование в области выявления англицизмов. В работе предложен улучшенный метод поиска английских заимствований в том числе с методами линейного отображения векторных пространств двух языков. Отличительной особенностью подхода является работа без подготовленных заранее словарей и собранных вручную коллекций.

Также рассматриваются вопросы распределения заимствований в различных корпусах русскоязычных пользователей социальных сетей. Предложена эффективная стратегия автоматического поиска текстовой информации, позволяющая уменьшить размер корпуса в 20 раз по сравнению со случайным сбором при сопоставимой полноте словаря. Сравнительный анализ материалов таких ресурсов как Живой Журнал, Вконтакте Твиттер показывает равномерность распределения заимствований в письменной речи пользователей различного пола и возраста.

Ключевые слова: поиск англицизмов, лексикография, дистрибутивная семантика, социально-сетевые тексты

1. Introduction

The widespread use of English in the process of globalization continues to have a tremendous impact on development of different languages, namely, the number of English words in them is growing rapidly. The phenomenon of Anglicisms is occurring in languages all over the world, and the Russian language is not an exception. In the field of natural language processing this tendency raises a problem, finding new words (loan words) that are not yet presented in dictionaries. The automatic detection of Cyrillic-written Anglicisms in Russian text is a new, non-trivial and actual problem, especially as it is representative of the texts of social networks. People commonly use loan words and orthographic variation of loan English borrowings in a significant way.

The notion of an Anglicism can be defined in various ways; what can be regarded as "true", as an Anglicism is a rather subjective issue. There are several types of English borrowings that we aim to detect:

- pure Anglicisms (*ex.: iPad—айпад, fashion—фэшн, YouTube—ютуб, etc.*)—the word written in Russian as it sounds in English;
- English roots, combined with Russian affixes (*ex.: gif+ка => гифка, от+football+ить => отфутболить, like+нуть => лайкнуть, etc.*)—the word has an English root and some Russian flexion;
- abbreviations (*ex.: LOL—лол, ZIP—зип, etc.*)
- composites (*ex.: life+hack—лайфхак, old+school—олдскул etc.*)—words with two English roots.

For the practical application of the proposed method it is important that a Russian word can be automatically linked to its English cognate.

A significant amount of theoretical works about integration of Anglicisms in Russian language, social-linguistic studies and interlanguage research are written (Chachibaia etc. 2005; Proshina, 2016; Janurik, 2010; Yaniv, 2016). The work (Chugunova etc. 2016) presents detailed classification of Anglicisms in Russian and continues the research of their adoption and origin. The authors (Muraviev, etc. 2014) study neologisms and loan words frequently occurring in Facebook user posts. The authors half-automatically collected a dataset of about 573 million posts from Russian-speaking users (written during the period from 2006 till 2013). As a result, authors produced a list of 168 neologisms, including Anglicisms and attempted to make etymological classification and distinguished thematic areas of these neologisms. Some research is devoted to classification of the modern Anglicisms on the Internet (Bylatcheva etc. 2016), others pay attention to the comparative studies of languages occupied by Anglicisms, as in the work made by Balakina (Balakina, 2011), where the author compares lexical items in Russian and German blogs. In one of his latest works (Dyakov, 2016) A. I. Dyakov classifies loan words and proposes an adaptation model of the Anglicism—the scheme of dynamic process in a Russian-speaker's thesaurus, frequency of use and mechanism of adoption. Over 10 years he manually collected more than 20,000 borrowed lexical items and from this considerable set of Anglicisms he created the Anglicism Dictionary1 available online.

For the Russian language, the method applicable to search for English loan words and their analogues in Russian social network texts was presented by authors of this paper (Fenogenova etc., 2016). The proposed general methodology on the material of LiveJournal texts was able to detect 1,146 Anglicisms based on 20 million LiveJournal texts and comments, but the proposed approach was limited by (a) the computational expense of machine translation procedure, proposed in the work, (b) the low fraction of Anglicisms in the collected corpus. Though the proposed method demonstrated relatively high recall, we failed to find many real-life Anglicisms due to their absence in our corpus. Thus, corpus formation (or network walk) strategy appears to be more of a vital problem rather than the hypotheses generation or the filtering strategy. Contributions to the present study are as follows:

- Modified Anglicism detection method: an approach to linear mapping between distributive vectors in different languages (Mikolov etc., 2013) was used instead of machine translation.
- Anglicism variety analysis: Anglicism distribution is independent to the data source and user age, sex, geographical location.
- Dictionary growth strategy: dictionary size is an asymptotic function of corpus size. The same amount of Anglicisms can be found on smaller corpus by means of effective data collection strategy.

[1] http://anglicismdictionary.dishman.ru/slovar

2. Anglicism Detection Algorithm

The method is based on the idea that the original Latin word is similar to its Cyrillic analogue in scripting, phonetics and semantics. We assume that words are likely to be borrowed if they sound or script in the same way as their English analogues. At the same time loan words and their original equivalents should be close in the distributional semantics model. From the corpus of social network texts we take words, mentioned more than 30 times and generate a list of hypotheses for each pair of words. Next, we make a list of possible transcriptions and transliterations from English words and compare them with Russian tokens by Levenshtein Distance. We get the Levenshtein Distance threshold as a function of word length, but the maximum threshold is set to 3 for the normal forms. As a result we get a list of hypotheses pairs and check them by distributional semantics in the following ways. The general architecture of our method is presented in picture 1.

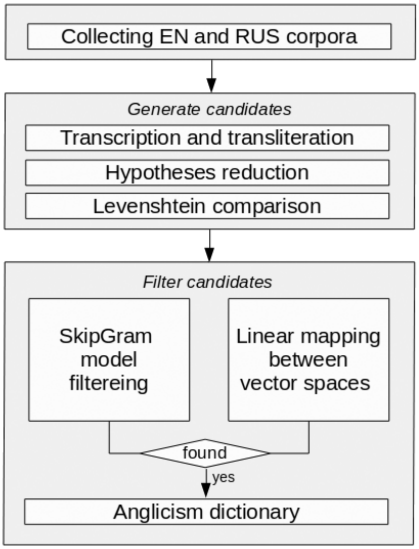

Collecting EN and RUS corpora
Generate candidates
Transcription and transliteration
Hypotheses reduction
Levenshtein comparison
Filter candidates
SkipGram model filtereing
Linear mapping between vector spaces
found
yes
Anglicism dictionary

Fig. 1. General algorithm

First, we verify our candidate from the hypotheses list appears in the model and has a Latin spelling as the most similar word that is equal to the English hypothesis candidate. Let us denote a hypotheses set as H, Anglicisms set as A. Any $h \in H$ consists of *h.rus*—a candidate to Anglicism, *h.eng*—prototype for Anglicism, *h.editDist*—Levenshtein edit distance between *h.rus* and *h.eng*. If *h.eng* in the top n nearest vectors, *h.rus* is proved to be Anglicism and we will form pairs (*h.rus*, *h.eng*) in set A.

Algorithm 1. Hypotheses validation

```
topByDist = {1000, 100, 10}
A=Ø
for all $h \in H$ do
    nearestVecs = w2vModel.getMostSimilar(h.rus)
  if h.eng in top topByDist[h.editDist] nearestVec then
    A.add((h.rus,h.eng))
 end if
end for
```

However, this method cannot cope with cases when the SkipGram model does not contain an English candidate. To solve the problem above, we trialed the method proposed by Mikolov (Mikolov etc., 2013) on our data. The English word2vec model was built and the linear mapping between vector space of English language and vector space of the Russian model was learnt. For linear mapping, we selected matrix *W* that minimizes

$$\sum_{i=1}^{m} \frac{1}{2} \|Wx_i - z_i\|^2,$$

where $x_i, z_i - i$—the pairing of an English word and its Russian translation. Next, mapping is provided between the linear vector that corresponded to the *hypothesis.rus,* and the vector of the word translation from the English vector space. The final step was to check the nearest top N vectors, if *hypothesis.eng* was proven to be in the list, *hypothesis.rus* was considered to be an Anglicism.

Algorithm 2. Hypotheses validation with translation

```
N = 100
A=Ø
for all $h \in H$ do
  vec = mapToRussianW2VSpace(h.eng)
  if h.rus in top N w2vModelRussian.nearestVec(vec) then
    A.add((h.rus, h.eng))
 end if
end for
```

3. Comparative experiments

For this study four datasets (LiveJournal, Twitter, Habrahabr and VKontakte) were used to investigate the distribution of Anglicisms. All selected online social networks have very wide topic coverage, user variety and ease of sampling a large dataset due to the public API. The source data, we used for training models and finding Anglicisms, is the following:

- VKontakte 11,426,003 Russian texts.
- Twitter 2,936,050 Russian texts.
- LiveJournal 10,000,000 Russian texts.
- Habrahabr 1,000,000 Russian texts.

To evaluate the proposed method we have used the following list of Anglicisms: we have combined the Dyakov dictionary with manually verified generated lists. The final dictionary contains 20,773 words. Subsequently, evaluation of our method will be performed based on this joint dictionary. The standard classification metric, F-measure, was used. It should be noted that due to the fact, that the algorithm cannot find Anglicisms that are missing in our corpus, we had to count only those words in the joint dictionary that have a frequency score of more than or equal to 30.

Table 1. Proposed method quality evaluated on different collections

Corpus	Method	True positive	False positive	Words of the joint dictionary in the corpus
VK+TW, LD ≤ 2	linear mapping	620	1,454	1,103
	SkipGram	323	235	
	linear mapping + SkipGram	823	1,638	
LJ, LD ≤ 2	linear mapping	506	323	4,321
	SkipGram	1,084	1,339	
	linear mapping + SkipGram	1,571	1,404	
Habr, LD ≤ 2	linear mapping	749	723	2,729
	SkipGram	534	139	
	linear mapping + SkipGram	1,060	554	

The corpus analysis on the material of blog-platform LiveJournal, that contains more than 20 million texts, has enabled us to detect Anglicisms. However, the intersection with the manually collected A. I. Dyakov dictionary of Anglicisms constitutes only 26%. At the same time more than 16,000 words were not presented in the LJ corpus at all. This allowed us to hypothesize—the Anglicism's usage was unevenly distributed among users of social networks. An alternative hypothesis was that users of LiveJournal did not use Anglicisms in their speech and writing at all. For hypothesis verification we have entailed statistical analysis of Anglicism distribution among users of VKontakte, Twitter, LiveJournal. additional analysis of user groups split by age and gender has been performed.

Distribution of Anglicisms among random users of VKontakte and LiveJournal social networks is shown in the picture 2 (a) and (b). Users who had at least 300 words in their texts were used to build the chart. The red color on the chart illustrates absolute number of Anglicisms in user texts. The blue color shows the number of unique Anglicisms. It shows that users from different social networks tend to apply Anglicisms in nearly the same proportion. Single Anglicisms were recognized between the 16th and the 25th word in user's speech irrespective of social network.

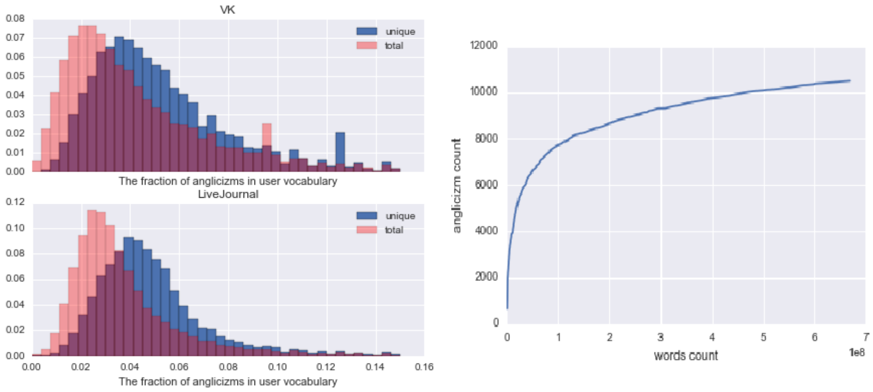


Fig. 2. The fraction of Anglicisms in a user's dictionary of **(a)** VKontakte **(b)** LiveJournal and the ratio of dictionary size to the size of the corpus with random search **(c)**

Frequency of Anglicism usage did not depend on gender, age or social network to be good evidence of fast adaptation of Anglicisms among users and their rapid integration into active dictionaries of online social network users researched. Analysis of VK user's age showed that users from age 12 to 35 tend to use new lexical items more actively than users from 40 to 70. The intersection between Anglicisms acquired from adults and teenagers was 0.62. Most frequent Anglicisms, used by one of the analyzed groups and almost never used by another is shown in table 2.

Table 2. Most frequent words, used only by teens and grown ups

Grown Up	Words frequency	Teen	Words frequency
маргарин (margarine)	1,602	шоурум (showroom)	957
бинго (bingo)	1,382	инст (inst)	311
компресс (compress)	1,009	мейк (make)	187
стак (stack)	969	трип (trip)	183
паприка (paprika)	851	спамить (spam)	158
форекс (forex)	729	хип-хоп (hip-hop)	149
майнинг (mining)	661	треш (trash)	147
компост (compost)	561	микроблейдинг (mikrobleyding)	147
тампон (tampon)	538	свитшот (svitshot)	140
тимьян (thyme)	510	кроссфит (crossfit)	139
рамблер (rambler)	492	пати (party)	137

Therefore, for actualization of dictionaries it's preferable to select young user's data, as Anglicisms used by people of the older generation are likely to be already contained in the dictionary. Furthermore, we can conclude that the hypothesis of uneven distribution of Anglicisms among Russian native speakers has been confirmed. Although observed social groups use different Anglicisms, the proportion of loan words in their speech is almost the same as it was in picture 2 (a) and (b). So we cannot say that teenagers use more Anglicisms, than grown-ups—observed words are different, but the proportion is the same.

Using the statistics of Anglicism usage, acquired from random user crawling, we have analyzed several dictionary formation strategies. Modelling Anglicisms by number of users that simultaneously use them, shows a strongly connected network with an average degree of 130. It assumes that we can significantly reduce a corpus size by focusing on users that have large amounts of Anglicisms in their speech. Our corpus formation algorithm had two steps, based on real crawling capabilities—(1) Get all texts (i.e. Anglicisms) of some user and (2) Get all users of some Anglicism. Step (1) supposes that we download all texts, include them to our corpus and proceed with the method described in section 2. To increase modelling speed, we made the assumption that Anglicisms can be found if they occur more than 30 times in the corpora. The number of Anglicisms can be estimated by multiplying the number of words by 0.35 to get the exact estimation (where 0.35 is the average F_1-measure quality of Anglicism detection by our method).

We took 100 users at each iteration; the user selection strategy was as follows:

- "Random"—select random users that were not included in to the previous iterations.
- "Rare A"—select users that actively use rare Anglicisms in their speech first.
- "Max A"—select users that use many Anglicisms in their speech first.
- "Max Lexic"—select users that have the richest vocabulary.

As we cannot see all user texts before we download them, we modeled our statistics on 100 randomly selected words written by this random user. This method simulates the situation when we observe comments associated with text already downloaded. We evaluated 1,000 experiments to get the mean statistic for each strategy. The resulting dictionary size ratio is shown in figure 3.

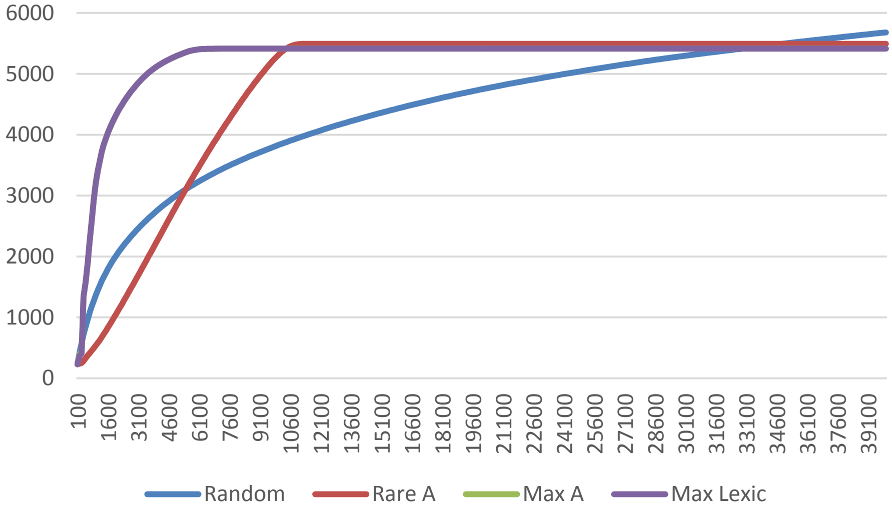


Fig. 3. The dictionary size to users downloaded ratio

As shown in figure 3, "Max A" and "Max Lexic" strategies give almost the same result. The dictionary size increases faster in the case of these strategies, although they are not able to get all Anglicisms found by random search strategy because some loan words stay separate from the rest of the vocabulary.

4. Conclusions

The following section breaks down three research contributions of this work and discusses their limitations. The linear mapping significantly increases total found borrowings recall and provides words missed by SkipGram model or naive translation. The resulting method is corpus dependent — it requires the same Russian word and its English analogue to be included into the corpus at least 30 times. The proposed method has satisfactory computational complexity that allows the researcher to verify hypotheses at Levenshtein Distance 2 or even more. Resulting recall at LD ≤ 2 is 0.74 that is significantly higher than all earlier observed results. The proposed method does not require precompiled dictionaries, however the use of the established dictionaries can be used to exclude old-fashioned Anglicisms and borrowings from other languages and giving researchers only contemporary, words unknown earlier.

Different social, gender, age groups, use different Anglicisms, although the percentage of loan words is nearly the same for all groups. Profile information should be used during the corpus formation as it increases the resulting Anglicism dictionary size. Teenagers use new lexical borrowings more actively than adult users, so the "New Anglicism Search" problem should be focused on a younger audience.

The best corpus formation strategy is the combination of random search and selection of rich vocabulary actors. First 5,000 users provide 95% of all Anglicisms contained in the corpus in this case.

Acknowledgement

The reported study is supported by RFBR grant № 16-29-09614 "Principles and technologies of machine processing for social network texts in Russian" and Russian Academic Excellence Project '5-100'.

References

1. *Balakina J.* (2011), Anglicisms in Russian and German Blogs, Frankfurt am Main: Peter Lang.
2. *Bylatcheva O. A., Safonkina O. S.* (2016), Internet Anglicisms: the development of English-Russian contacts today [Internet-anglicismy: rasvitie anglo-russkich yazykovych kontaktov na sovremennom etape] //Ogariov-Online, № 17 (82).
3. *Chachibaia N. G., Colenso M. R.* (2005), New Anglicisms in Russia, In and Out of.
4. *Chugunova E. I., Runtova N. V.* (2016), The origin and classification of Anglicisms in Russian language [Proishozhdenie i klassifikazia anglicismov v russkom yazyke], The modern tendencies of development of science and technology [Sovremennye tendencii rasvitiya nauki i technologij], № 10-6, pp. 135–138.

5. *Dyakov A. I.* (2016), Adaptational model of Anglicisms [Adaptazionnaya model anglicismov], Scientific researches: from theory to practice [nauchnye issledovania: ot teorii k praktike], № 3 (9), pp 245–255.
6. *Fenogenova A., Karpov I., Kazorin V. A.* (2016), General Method Applicable to the Search for Anglicisms in Russian Social Network Texts., AINL-FRUCT, IEEE p. 1–6.
7. *Gisle Andersen* (2005), Assessing algorithms for automatic extraction of Anglicisms in Norwegian texts, Corpus Linguistics.
8. *Janurik S.* (2010), The integration of English loanwords in Russian: An overview of recent borrowings, Studia Slavica, T. 55, № 1.,pp. 45–65.
9. *Kristiansen M.* (2013) Detecting specialised neologisms in researchers' blogs, Bergen Language and Linguistics Studies, T. 3, № 1.
10. *Leidig S., Schlippe T., Schultz T.* (2014), Automatic detection of Anglicisms for the pronunciation dictionary generation: a case study on our German IT corpus, SLTU, pp. 207–214.
11. *Mikolov T., Le Q. V., Sutskever I.* (2013) Exploiting similarities among languages for machine translation; preprint arXiv:1309.4168.
12. *Muraviev N. A., Panchenko A. I., and Obiedkov S. A.* (2014), Neologisms on Facebook, Dialog.
13. *Proshina Z.* (2008) English as a Lingua franca in Russia, Intercultural Communication Studies, T. 17., № 4., pp. 125–140.
14. *Serigos J.* (2016), Using distributional semantics in loanword research: A concept-based approach to quantifying semantic specificity of Anglicisms in Spanish, International Journal of Bilingualism, pp. 1367006916635836.
15. *Yaniv O.* (2016), Anglicisms in the Russian Language Based on-ing Borrowings.

LEARNING NOISY DISCOURSE TREES

Galitsky B. (boris.galitsky@oracle.com)

Oracle Corp Redwood Shores CA USA

It is well known that syntax-level analysis of user-generated text such as tweets and forum postings is unreliable due to its poor grammar and incompleteness. We attempt to apply a higher level linguistic analysis of rhetoric structure and investigate the potential application domains. We leverage an observation that discourse-level structure can be extracted from noisy text with higher reliability than syntactic links and named entities. As noisy text frequently includes informal interaction between agents, discussions, negotiations, arguments, complaints, we augment discourse trees with speech acts. Speech Act discourse tree (SADT) is defined as a discourse tree with verbs for speech acts as labels for its arcs. We identify text classification tasks which relies on tree kernel learning of SADTs: detection of negative mood (sentiment), text authenticity and answer appropriateness for question answering in social domains. The results are that the proposed technique outperforms on the discourse level traditional keyword-based algorithms in all of these three tasks.

Keywords: Speech Act Discourse Tree, sentiment analysis, answer appropriateness, text authenticity

1. Introduction

It is well known that text related to social network domains, what is called a user-generated data, is noisy. Therefore application of traditional natural language methods to texts written by non-professionals gives lower accuracy. One can expect that while processing noisy data (Jørgensen et al. 2015), certain level of generalization and abstraction would be beneficial. Similarly to other media such as images, an ascent to a higher-level of analysis would be fruitful.

In the last couple of years, availability of parsers which produce discourse structure significantly improved. Discourse parsers allows for an efficient automated analysis of rhetoric structures of text (Webber 2012, Joty et al 2013, Feng and Hirst 2014, Surdeanu et al 2015). Accuracy of discourse representations of rhetoric parsers has significantly improved, so that obtained discourse trees can be a subject of further automated analysis. However, a corpus of studies on applications of computed discourse trees is rather limited. In this study we explore how high-level discourse analysis of noisy text can be leveraged by a number of applications where traditional NLP techniques are fairly limited.

Marcu (1998) regarded a document as a Rhetorical Structure Theory (RST) (Mann & Thompson, 1988)-based discourse tree and selected textual units according to a preference ranking derived from the tree structure to make a summary. Representing an essence of a noisy text can be viewed from the text summarization perspective. Recent studies on text summarization formulate it as a combinatorial optimization problem, extracting the optimal subset from a set of the textual units that maximizes an objective function without violating the length constraint. Although these methods successfully improve automatic evaluation scores, they do not consider the discourse structure in the source document. To be logically coherent, (Hirao et al 2015) proposed a method that exploits a discourse tree structure to produce coherent summaries, transforming a traditional discourse tree, namely a rhetorical structure theory-based discourse tree, into a dependency-based discourse tree.

Chat bots frequently rely on ad-hoc solutions for the units making chat turn decisions (Popescu 2007). However, with the advent of novel dialogue planning techniques, integrating task-specific and general world knowledge in order to provide a more reliable and natural interaction with humans, more sophisticated chatbot response generation techniques are necessary. The authors present performance improvements employing a module that simplemented Segmented Discourse Representation Theory for response generation for chatbots, using the first-order logic (FOL) formalism, enforced by a task-independent discourse ontology. These improvements concern reductions in computational costs and enhancements in rhetorical coherence for the discourse structures obtained, and are obtained using speech-act related information for driving rhetorical relations computations.

Although discourse parsers rely on syntactic information, we expect them to perform reasonably well even when this information such as part-of-speech tags and syntactic trees are incomplete and noisy (van der Wees et al 2015). To further overcome this noisiness problem, we extend discourse trees with speech acts extracted from text to better represent the structure of what noisy text authors communicate and in which way.

Notice that slightly different texts might produce rather different DTs, and conversely, totally unrelated texts can produce the same DT even if the flow of rhetoric relations is not fully identical. DT parsers produce differences, which are not necessarily anchored in true discourse facts. Speech Act-based discourse trees (SADT) help to overcome this problem, since the traditional DTs are enriched with communicative discourse so that even if relations are misrepresented due to falsely fired syntactic rules, the structure of communication is still retained. We combine Speech Act Theory (Searle 1969) and Rhetoric Structure Theory. SADTs, in addition to relation between fragments of texts connected with rhetoric relations (Mann et al 1992), have special labels related to speech acts used by participants of a scenario to present a given rhetoric relation to the reader of the text.

Noisy discourse tree appear in the following tasks:

- Detecting a *logical argument* in text. A number of text genres of noisy text include argumentation, where an author attempts to back up her claim with certain statement. Argumentation is frequently associated with heated discussion. Conversely, multiple genres such as fact sharing, instructions and others do not include argumentation. In a content management system, it is important to automatically relate a noisy text to either the class of opinionated texts (with argumentation) or to the unbiased class (without argumentation).

- *Sentiment analysis*. Traditional, semantic compositionality methods of sentiment analysis are unreliable even when the text is not noisy. A user mood such as negative sentiment can be inferred from such paragraph-level features as intense argumentation, complex mental states such as deception, and others. Customer reviews and opinionated text is a good source of noisy text to explore how sentiment polarity can be inferred from the discourse-level features, since the lower-level linguistic features are rather noisy and unreliable.
- Text authenticity (validity, soundness, proper communication, confidence). It is rather harder to assess style-related features of a grammatically incorrect text based on its syntactic features. The degree of grammar deviation from normal is not a good indicator of content validity. It is hard to form explicit rules for how text style corresponds to its validity, therefore a supervised learning approach seems to be more plausible. An interesting and systematic example here are customer complaints, where the task of a customer support agent is to differentiate between
    1) valid, sound complaints requiring attention, from
    2) invalid, fake ones where a user is in a bad mood or just intends to receive a compensation.
- *Answer appropriateness commenting on a user post.* This is a special case of question answering, an automated support of user conversation, where the seed (the question or a request) is an incomplete or grammatically incorrect paragraph of text. To support a dialogue, a conversational agent needs to extract a topic from a seed and also maintain the coordination between the seed and response.

In all these domains, the problem is formulated as text classification into two classes:

- Positive (sentiment, authentic / valid text, correct answer or reply);
- Negative (sentiment, incorrect / invalid / incohesive text, incorrect answer or reply).

For a text to be classified into one of these classes, it has to be similar to its elements. We use statistical learning of structures with implicit feature engineering in the form of kernel learning of discourse trees as a reduction of a set of parse trees for a paragraph. If the solution to these problems for noisy text is satisfactory we can expect a broader range of application based on SADTs.

It turns out that using only rhetoric relations or only speech acts gives insufficient accuracy, but the combination of these sources produce acceptable results. More detailed syntactic and discourse information might help but can be redundant as well. In this study we will rely on information obtained from rhetoric relations and speech acts and compare the results with a classification system employing the syntactic data only.

If a text is shorter than a paragraph, such as Twitter, discourse-level analysis is believed to be inappropriate.

2. Representing a purpose of text in its DT

Conducting content exploration via chat bots or search engines (Galitsky 2013), discourse analysis is expected to help shortlisting answers are coordinated with a question in terms of style. The way an answer is communicated should be coordinated with the way a question is formulated. For example, if a user asking a question is a specialist in a certain adjucent area, an answer should contain a link between this specialty area and the focus of the question. The role of achieving agreement between user questions and user answers is especially high in noisy text domain.

Discourse-level agreement demands that A matches Q with respect to a domain knowledge and confidence, argumentation style, a level of politeness and other text features other than topics. On the other hand, discourse-level considerations are applied to Q/A topicality as well. If Q is represented as a sequence of keywords, and A is represented as a DT, then it is possible to formulate a simple rule-based system to filter out irrelevant answers based on how query keywords are distributed through the DT-A. These rules can be considered as constraints for the mapping between the nodes of trees

DT-Q → DT-A,

where PT(Q) is a trivial tree, a chain of words (we remove all edges and add unlabeled edges to link the nodes for words in a sequence); and DT-A is a tree with nodes for words and edges for rhetoric relations (all other edges are removed). Once an answer text is split into elementary discourse units (EDUs), and rhetoric relations are established between them, we establish rules for whether query keywords occurring in text are connected by rhetoric relations (and therefore this answer is likely relevant) or not connected (and this answer is most likely irrelevant). Hence we use a discourse tree (DT) as a base to identify certain sets of nodes in the DT to corresponding to Qs so that this text A is a valid answer, and certain sets of nodes correspond to invalid answers.

Usually, the main clause of a multi-sentence question includes the main entity of Q and some of its attributes, and supplementary clauses include other attributes and possibly constraints on them. In the most straight-forward way, the main clause of a question is mapped into a nucleus, and the supplementary clause is mapped into a satellite of the RST relation, such as elaboration. Linkage by other RST relations, where a satellite introduces additional constraints for a nucleus, has the same meaning for answer validity. This validity still holds where two EDUs are connected with a symmetric relation, such as joint. However, when the images of the main and supplementary clause of Q are satellites of *different* nuclei in A, it most likely means that they express constraints for different entities and therefore this A is irrelevant for this Q.

We start with an example of answer text split into EDUs:

> [furthermore,] e1 [they think] e2 [stock volatility maximum is not occurring at the same time in the past,] e3 [because of production and pricing differences] e4 [that are limiting the accuracy of seasonal adjustments] e5 [built into the financial data.] e6

A DT including 6 nodes {e1…e6} is shown in Fig. 0. Horizontal lines indicate text segments; satellites are connected to their nuclei by curved arrows.

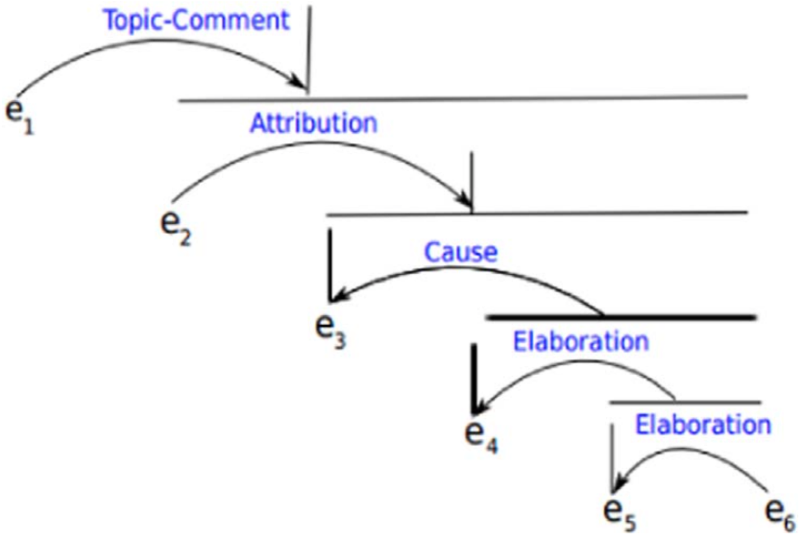


Fig. 0. Initial example of a DT

One can see that this text is a relevant answer for the question

Are seasonal swings in stock price volatility due to pricing differences?

because the respective areas e3 and e4 in the DT-A are ({*stock, volatility, maximum, …, due, pricing, differences*}) → DT-Q({… e3, e4, …}), {*seasonal, swings*} → e3, {*pricing, differences*} → e4. However, this answer is irrelevant for the question

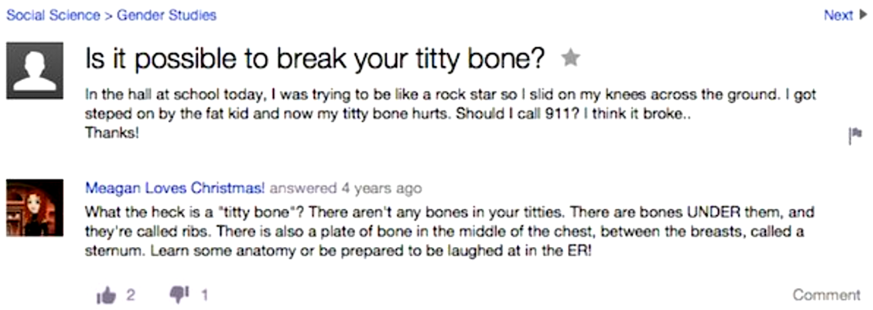
Are pricing differences built into employment data?

because the areas e4 and e6 in the rhetoric map of the answer are not connected. EDU e6 is an elaboration of e5, and e5 is, in turn, an elaboration of e4; however, e4 and e6 are not logically connected and cannot be mapped into by a set of question keywords.

3. Mapping DT-Q into DT-A

We introduce an example of a question and its answer (CollegeHumor 2017) and show that their DTs have to agree. We will demonstrate that SDT is an andequate means to express this form of agreement. If a question has a certain logic expressed by a discourde structure, the answer has to match it in some way. Q/A pair and the respective pair of discourse trees is shown in Fig. 1 (Q is on the left and A is on the right).

The main contradiction in this Q/A pair is that the Q demonstrates a lack of knowledge on a subject and A includes an argument that this knowledge needs to be acquired. Relation *contrast* in Q has to be addressed in A. Since Q is asking whether an accident is serious (and a trip to emergency room is necessary) or not, A has to include this relation, considering cases when the Q author is knowledgable in anatomy or not and how it affects the emergency room visit. Hence we map *Q-contrast* into *A-contrast*. Also, the *elaboration* relation associated with *Q-contrast* is mapped into *elaboration* relation associated with *A-contrast*.



Is it possible to break your titty bone?

In the hall at school today, I was trying to be like a rock star so I slid on my knees across the ground. I got steped on by the fat kid and now my titty bone hurts. Should I call 911? I think it broke..
Thanks!

Meagan Loves Christmas! answered 4 years ago

What the heck is a "titty bone"? There aren't any bones in your titties. There are bones UNDER them, and they're called ribs. There is also a plate of bone in the middle of the chest, between the breasts, called a sternum. Learn some anatomy or be prepared to be laughed at in the ER!

As to a formal definition of a SADT, it is as follows. SADT is a DT with labels for arcs that are the VerbNet expressions for verbs which are related to speech acts. The arguments of these verbs are substituted from text according to VerbNet frames. The first argument is instantiated by an agent and the second by a noun phrase that is a subject of a speech act. Further details on DTs are available in (Joty et al 2016), and on VerbNet Frames—in (Kipper et al 2008).

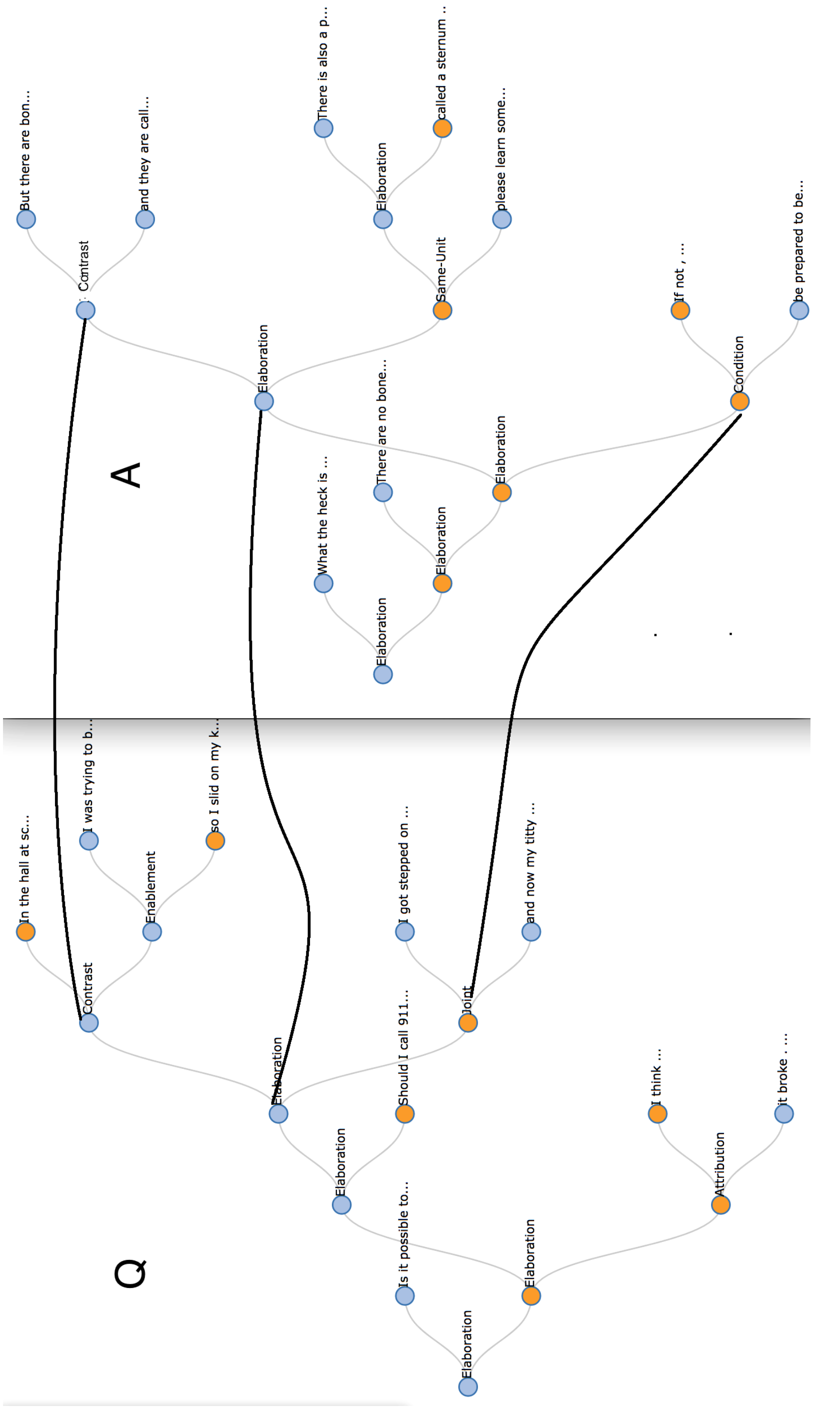


Fig. 1. Discourse Trees for a question and an answer have to be coordinated

4. Similarity function for learning SADT

Deep learning approach is not well suited to be applied to structured data since feature engineering and explainability are difficult. Deep learning can potentially apply a more complex feature space and assure a higher classification accuracy, but does not help in understanding or exploring the phenomena. We therefore use inductive and statistical approaches:

1) Represent SADTs in a numerical space, and express similarity as a number. This is a tree-kernel approach that belongs to statistical learning family. The feature space includes all SADTs sub-trees.
2) Use a structural representation, without numerical space, such as trees and graphs, and express similarity as a maximal common sub-structure (Galitsky 2012, 2016). We refer to such operation as *similarity operation* (*generalization*, '^'). This is an inductive learning approach.

We use the former approach to assess how text classification tasks can be consistently handled when the data becomes noisier and syntactic analysis produces more errors. The latter one is superior in terms of feature engineering but is also less universal and would need special representations of DTs depending on an application area. Therefore a hybrid approach combing best of both worlds would be beneficial (not evaluated in this study).

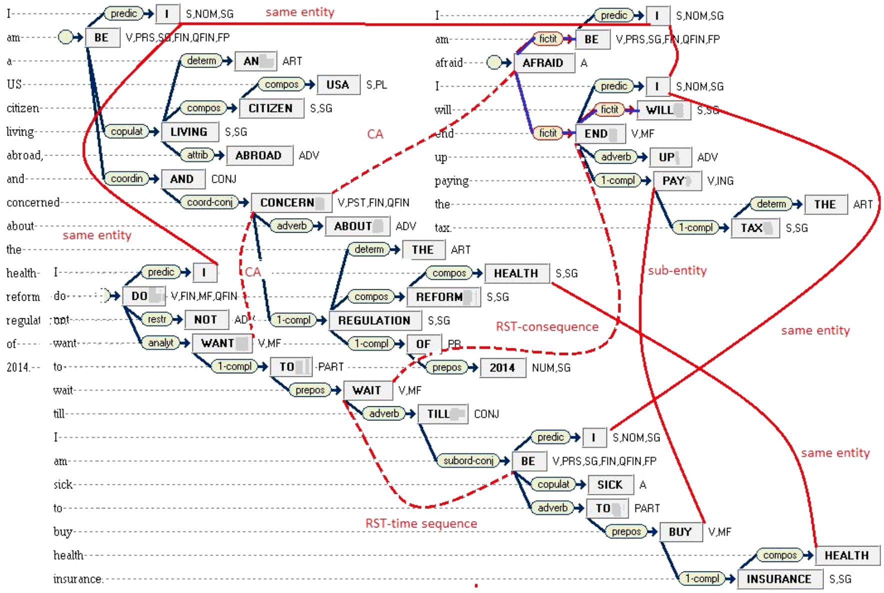

I am a US citizen living abroad, and concerned about the health reform regulat of 2014.
I do not want to wait till I am sick to buy health insurance.
I am afraid I will end up paying the tax.
same entity
same entity
CA
CA
sub-entity
RST-consequence
same entity
same entity
RST-time sequence

Fig. 2: A Parse thicket for a question

(Galitsky et al 2015) combined parse trees for sentences with discourse-level relationships between words and parts of the sentence in one graph, called *parse thicket*. The straight edges of this graph are syntactic relations, and curvy arcs—discourse relations, such as anaphora, same entity, sub-entity, rhetoric relation and communicative actions. Fig. 2 shows the parse thicket for the question *I am a US citizen living abroad, and concerned about the health reform regulation of 2014. I do not want to wait till I am sick to buy health insurance. I am afraid I will end up paying the tax.*

Parse thicket includes much more complete information than just a combination of parse trees for individual sentences would, especially when these trees are noisy. Navigation through the parse thicket along the edges for syntactic relations as well as the arcs for discourse relations allows one to transform a given parse thicket into semantically equivalent forms for matching with other parse thickets, performing a text similarity assessment task at the level of paragraph, irrespectively how it is split into sentences. Parse thickets also help to do relevance assessment with noisy text where syntactic analysis is subject to numerous errors and omissions. SADT is a subtree of parse thicket as a graph with the focus on rhetoric-level information only.

4.1. Tree Kernel learning for SADT

Tree Kernel learning for strings, parse trees and parse thickets is a well-established research area nowadays. The parse tree kernel counts the number of common sub-trees as the discourse similarity measure between two SADTs. Tree kernel has been defined for DT by (Joty and Moschitti 2014). (Wang et al 2013) used the special form of tree kernels for discourse relation recognition. In this study we extend the tree kernel definition for the SADT, augmenting DT kernel by the information on communicative actions. A SADT can be represented by a vector of integer counts of each sub-tree type (without taking into account its ancestors).

We combined Stanford NLP parsing, coreferences, entity extraction, DT construction (discourse parser, Surdeanu et al 2013 and Joty et al 2016), VerbNet and Tree Kernel builder into one system available at https://github.com/bgalitsky/relevance-based-on-parse-trees.

5. Evaluation of SVM TK learning of SADT in four domains

To detect sentiments, we first need to learn to detect a mixture of opinions, a conflict, a presence of logical argumentation in text. Then we build a hybrid sentiment classification system relying upon the detected cases of opposing argumentation.

5.1. Detecting noisy argumentation

We formed the *positive* dataset from the noisy text data where argumentation is frequent, e.g. opinionated letters to the editors of major US newspapers. We also used textual customer complaints dataset from our previous evaluations. Besides, we use the text style & genre recognition dataset (Lee, 2001) which has a specific dimension associated with argumentation. For the *negative* dataset, we used a non-noisy text sources such as Wikipedia and factual news sources. Both datasets include 3,600 texts.

Table 1: Evaluation results for detecting logical argument

Method / sources	Newspaper opinions			Customer complaints		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Naïve Bag-of-words	63.4	56.7	59.86	52.3	54.2	53.23
WEKA-Naïve Bayes	64.7	57	60.61	56.7	52.6	54.57
SVM TK for RST and SA (full parse trees combined in parse thicket)	78.8	72.9	75.74	74.6	70.2	72.33
SVM TK for DT (w/o SA)	62.4	61.7	62.05	59.3	63.2	61.19
SVM TK for SADT	81.9	76.3	79.00	75.2	74.6	74.90

SVM TK baseline is shown as light-greyed area in the middle row of Table 1. Representation includes exhaustive syntactic information in the form of parse thickets. The best algorithm of the current study, SVM TK for SADT (bottom greyed row) outperforms SVM TK for traditional DTs (without speech acts) by as much as 25% and full-set syntactic features (the SVM TK baseline) by only 3%. We conclude that contribution of speech act—related information for noisy text is substantial. A small gain in accuracy is due to the fact that noisy text syntactic data is noisy, and its addition decreases the recognition accuracy instead of increasing it.

5.2. Improvement of sentiment detection

Since reliable sentiment detection in an arbitrary domain is extremely hard, we focus on a particular sentiment—related feature such as logical argumentation and observe how its detection (Section 4.1) can help overall sentiment assessment. We formulate sentiment detection problem for noisy text at the level of paragraphs, only detecting sentiment polarity. For evaluation, we use a dataset of positive and negative, genuine and fake travelers review of Chicago area hotels (Ott et al 2013).

The results of sentiment analysis achieved by the hybrid compositional semantics and discourse analysis are shown in Table 2. In the first row we show the accuracy of the baseline system on our data. In the second grayed row we show the improvement by means of the hybrid system. This improvement of almost 15% is achieved by discovering overall negative sentiment at the paragraph level in case of recognized presence of argumentation. In some of these cases the negative sentiment is implicit and can only be detected indirectly from the discourse structure, where individual words do not indicate negative sentiments.

Table 2. Evaluation of sentiment analysis task

Data source and method	Precision	Recall	F-measure
Baseline sentiment detector (Standord NLP Sentiment)	62.7	68.3	65.38
Hybrid sentiment detector (Stanford NLP + SVM TK for SADT)	79.3	81.0	80.14
Sentiment detector via SVM TK for SADT	69.8	68.3	69.04

5.3. Accessing authenticity of customer complaints and reviews

Table 3. Evaluation of complaint/review validity task

Data source and method	F
Untruthful opinion data detector, *positive* reviews (SVM TK SADT)	77.26
Untruthful opinion data detector, *negative* reviews (SVM TK for SADT)	76.23
SVM TK of unconnected parse trees	62.84
SVM TK of parse thicket with anaphora only	65.10
SVM TK of with anaphora and Stanford sentiment profiles	74.46
SVM TK of parse thickets with anaphora and RST	78.98
SVM TK of SADT	80.03

We explored whether fake opinionated text have different rhetoric structure to genuine one (Table 3).

Although our SVM TK system did not achieve (Ott et al 2011, 2013) performance of 90%, the task of detection of fake review texts was performed (at 76–77% accuracy, two bottom greyed rows) by the universal text classification system, the same that extracts arguments, finds rhetorically suitable answers and assesses sentiments polarity. We also accessed the validity of customer complaints, based on the manually tagged set of a limited size. We observed how adding discourse information improves recognitions accuracy: we start with unconnected parse trees, then add anaphora and RST, and finally proceed to SADT (bottom of Table 3).

5.4. Assessing coordination a question and an answer

Our evaluation dataset included 560 Answers and Questions scraped from public sources. We consider the pair *Question-Best Answer* as an element of the positive training set and *Question-OtherAnswer* as the one of the negative training set.

To facilitate data collection, we designed a crawler which searched a specific set of sites, downloaded web pages, extracted candidate text and verified that it is adhered to a question-or-request vs response format. Then the respective Q/A pair of texts is formed. The search is implemented via Bing Azure Search Engine API in the Web and News domains.

Answer classification accuracies are shown in Table 4. Each row represents a particular method; each class of methods in shown in grayed areas.

Table 4. Evaluation of the coordination task

	Community Answers		
Source / Evaluation setting	P	R	F1
Types and Counts for rhetoric relations of Q and A	55.2	52.9	54.03
Entity-based alignment of DT of Q and A	63.1	57.8	60.33
SVM TK for Parse Trees of individual sentences	66.1	63.8	64.93
SVM TK for RST and SA (parse thickets	75.8	74.2	74.99

Source / Evaluation setting	Community Answers		
	P	R	F1
SVM TK for RR-DT	76.5	77.0	76.75
SVM TK for RR-SADT	80.3	78.3	79.29
SVM TK for RR-SADT + sentiment + argumentation features	78.3	76.9	77.59

Our evaluation settings are close to SVM-based ranking of RST parses. The rhetoric relevance recognition accuracy is also comparable with the state of art in question answering systems relying on rhetoric features such as (Jansen et al. 2013).

Conclusion

In this study we defined SADT and proposed a statistical SVM TK based learning framework that can be applied to a manifold of NLP tasks. SADT allows combining the structure of rhetoric relation with the structure of communication, which complements each other being applied to noisy noisy text.

Using SVM TK one can differentiate between a broad range of styles of noisy text (Galitsky et al 2015). Each text style and genre has its inherent rhetoric structure that is leveraged and automatically learned. When syntactic structure is noisy and some features can be missing, the rhetoric structure with unreliably detected EDUs can still be a reliable indicator of text style. Since the correlation between text style and text vocabulary is rather low, traditional classification approaches, which only take into account keyword statistics information could lack the accuracy in the complex cases.

An extensive corpus of literature on RST parsers does not address the issue of how the resultant DT will be employed in practical NLP systems. RST parsers are mostly evaluated with respect to agreement with the test set annotated by humans rather than its expressiveness of the features of interest. In this work we focused on interpretation of DT for noisy text and explored ways to represent them in a form indicative of a conflict, negative sentiment, sharing of authentic information rather than neutral enumeration of facts.

We demonstrated that this discourse-level technique performs better than traditional keyword-based statistical and/or compositional semantics approaches in all of these four tasks. We also showed that this improvement is larger for user-generated content in comparison with the professionally written text with proper style and grammar. Classification of SADTs gives a higher accuracy than a conventional sentiment analysis. Text validity assessment for a gives satisfactory results, comparable to general style classification accuracies obtained elsewhere. Also, rhetoric support for answer relevance demonstrated the accuracies comparable with the state-of-the-art for community question answering (Jansen et al 2013).

The code used in this study is open source and is available at https://github.com/bgalitsky/relevance-based-on-parse-trees.

References

1. *Lee, D.* (2001) Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the BNC jungle. Language Learning & Technology, Vol. 5(3): 37–72.
2. *Collegehumor.com* (2017). http://www.collegehumor.com/post/7013323/22-yahoo-answers-questions-that-just-might-rot-your-brain/page:3. Last downloaded Feb 10, 2017.
3. *Mann William and Sandra Thompson.* (1988) Rhetorical structure theory: Towards a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.
4. *Webber, B., Mark Egg, and Valia Kordoni.* (2012) Discourse structure and language technology. Natural Language Engineering, 18:437–490.
5. *Marcu, D.* (1998) Improving summarization through rhetorical parsing tuning. In Proc. of the 6th Workshop on Very Large Corpora, pages 206–215.
6. *WenTing Wang, Su Jian, Chew Lim Tan.* (2010). Kernel Based Discourse Relation Recognition with Temporal Ordering Information. ACL.
7. *Feng, Vanessa Wei and Graeme Hirst.* (2014) A linear- time bottom-up discourse parser with constraints and post-editing. In Proceedings of The 52nd An- nual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, USA, June.
8. *Joty, S., G. Carenini, R. T. Ng, and Y. Mehdad.* (2013) Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In ACL (1), pages 486–496.
9. *Joty, S. and A. Moschitti.* (2014) Discriminative Reranking of Discourse Parses Using Tree Kernels. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).
10. *Joty, Shafiq R., Giuseppe Carenini, Raymond T Ng.* (2016) CODRA: A Novel Discriminative Framework for Rhetorical Analysis. Computational Linguistics Volume 41, Number 3, 2016.
11. *Jørgensen, Anna, Dirk Hovy and Anders Søgaard.* (2015) Challenges of studying and processing dialects in social media. Proceedings of the ACL 2015 Workshop on Noisy User-generated Text.
12. *P. Jansen, M. Surdeanu, and P. Clark.* (2014) Discourse Complements Lexical Semantics for Nonfactoid Answer Reranking. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL).
13. *Galitsky B., Ilvovsky D., Kuznetsov S. O.* (2015) Text Classification into Abstract Classes Based on Discourse Structure, in: Proceedings of the Recent Advances in Natural Language Processing, RANLP 2015. Hissar. P. 201–207.
14. *Galitsky, B.* (2012) Machine Learning of Syntactic Parse Trees for Search and Classification of Text. Engineering Application of AI.
15. *Galitsky, B.* (2016) Generalization of parse trees for iterative taxonomy learning. Information Sciences. Volume 329, n1 February 2016, Pages 125–143.
16. *Galitsky B.* (2013) Content inversion for user searches and product recommendations systems and methods. US Patent 9336297. https://www.google.com/patents/US9336297.

17. *Kipper K., Korhonen A., Ryant N. and Palmer M.* (2008) A large-scale classification of English verbs. Language Resources and Evaluation Journal, 42, pp.21–40.
18. *Popescu, Vladimir, Jean Caelen, Corneliu Burileanu* (2007) Using Speech Acts in Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue. Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3–7, 2007, Proceedings
19. *Searle, J.* (1969). Speech acts: An essay in the philosophy of language. Cambridge: Cambridge University Press.
20. *Surdeanu, Mihai, Thomas Hicks, and Marco A. Valenzuela-Escarcega.* (2015) Two Practical Rhetorical Structure Theory Parsers. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics—Human Language Technologies: Software Demonstrations (NAACL HLT).
21. *Ott, M., Y. Choi, C. Cardie, and J. T. Hancock.* (2011) Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.
22. *Ott, M., C. Cardie, and J. T. Hancock.* (2013) Negative Deceptive Opinion Spam. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.
23. *Jindal, N., and Liu, B.* (2008) Opinion spam and analysis. In WSDM, 219–230.
24. *Marlies van der Wees, Arianna Bisazza and Christof Monz.* (2015) Five Shades of Noise: Analyzing Machine Translation Errors in User-Generated Text. Proceedings of the ACL 2015 Workshop on Noisy User-generated Text.
25. *Hirao, T., Masaaki Nishino, Yasuhisa Yoshida, Jun Suzuki, Norihito Yasuda, and Masaaki Nagata.* (2015) Summarizing a document by trimming the discourse tree. IEEE/ACM Trans. Audio, Speech and Lang. Proc. 23, 11 (November 2015), 2081–2092. DOI=http://dx.doi.org/10.1109/TASLP.2015.2465150

COMPLEX APPROACH TOWARDS ALGORITM LEARNING FOR ANAPHORA RESOLUTION IN RUSSIAN LANGUAGE

Gureenkova O. A. (ol.gure@gmail.com)[1],
Batura T. V. (tatiana.v.batura@gmail.com)[2,3],
Kozlova A. A. (noriel266@gmail.com)[2],
Svischev A. N. (alekseisvischev@gmail.com)[2]

[1]Expasoft Ltd., Novosibirsk, Russia; [2]Novosibirsk State University, Novosibirsk, Russia; [3]A. P. Ershov Institute of Informatics systems, Novosibirsk, Russia

The paper considers applying of ensemble algorithm based on rules and machine learning for anaphora resolution in Russian language. Ensemble presents combination of formal rules, a machine learning algorithm Extra Trees and an algorithm for working with imbalanced learning sets Balance Cascade. Complexity of the approach lies in generation of complex features from rules and vectorization of syntactic context, with context data obtained from algorithms mystem (Yandex), SyntaxNet (Google) and Word2Vec.

Key words: anaphora, antecedent, cataphora, Random Forest, machine learning, imbalanced set, Extra Trees, Balance Cascade, SyntaxNet, Word2Vec

КОМПЛЕКСНЫЙ ПОДХОД К ОБУЧЕНИЮ АЛГОРИТМОВ ДЛЯ РАЗРЕШЕНИЯ АНАФОРЫ В РУССКОМ ЯЗЫКЕ

Гуреенкова О. А. (ol.gure@gmail.com)[1],
Батура Т. В. (tatiana.v.batura@gmail.com)[2,3],
Козлова А. А. (noriel266@gmail.com)[2],
Свищев А. Н. (alekseisvischev@gmail.com)[2]

[1]ООО «Экспасофт», Новосибирск, Россия; [2]Новосибирский государственный университет, Новосибирск, Россия; [3]Институт систем информатики им. А. П. Ершова СО РАН, Новосибирск, Россия

В работе рассматривается применение ансамбля алгоритмов, основанного на правилах и машинном обучении для разрешения анафоры в русском языке. Ансамбль представляет собой объединение формальных правил, алгоритма машинного обучения Extra Trees и алгоритма для работы с несбалансированной выборкой Balance Cascade. Комплексность подхода заключается в генерации сложных признаков, полученных на основе правил и векторизации синтаксического контекста с учетом данных из алгоритмов mystem (Yandex), SyntaxNet (Google) и Word2Vec.

Ключевые слова: анафора, антецедент, катафора, случайный лес, машинное обучение, несбалансированная выборка, Extra Trees, Balance Cascade, SyntaxNet, Word2Vec

Introduction

In such natural language processing tasks as machine translation, information extraction and others the engineers often face the problem of anaphora resolution. Resolution of personal pronoun anaphora is the task of finding a word expression that a personal pronoun refers to. There is a significant number of researches related to anaphora resolution in the European languages [2, 3, 5]. As for the Russian language, the problem is not sufficiently represented. This is due to the fact that there is a lack of open annotated Russian corpora that are required for model training and evaluation.

The basic concepts related to the task of pronominal anaphora resolution are anaphor and antecedent. Consider the example: "***Человек** ленив по своей природе, и только жесточайшая конкуренция может привести **его** к успеху*." The word "*его*", anaphor, refers to the same real-world entity that "*человек*", antecedent. Commonly, the antecedent is located in the text before the anaphor. But there are also cases of *cataphora*—a phenomenon, opposite to anaphora, when the antecedent appears in the text after the pronoun. In this work we aggregate these two terms into the one—anaphora.

The aim of our work was to develop an algorithm that resolves anaphoric links between the personal pronouns and their antecedents in the Russian language. In this algorithm we used a hybrid approach, based on rules and machine learning.

The anaphora resolution in Russian involves certain difficulties. Usually text preprocessing includes the following steps: part-of-speech tagging, morphological analysis of words, detection of noun phrases, syntactic parsing and surface-semantic analysis. During automatic preprocessing the errors tend to appear and accumulate at the following steps, and later the errors may affect the algorithm quality. However, interest in the problem of anaphora resolution remains high in recent years among both European and Russian researchers.

1. Related works

We can distinguish three approaches to anaphora resolution: rule-based, based on machine learning and hybrid.

A rule-based approach is suggested in [3]. The main idea is to take into account, besides part-of-speech tagging, the information about the noun phrases preceding the anaphora at the distance of two sentences. Only those noun phrases that agree with the anaphora in gender and number are selected. Then the rules are applied consistently. The size of text corpus for testing the algorithm was 28 thousand words (among which 422 pronouns). The accuracy was 57%.

The paper [1] describes a rule-based method of anaphoric links detection for the Russian language. The research is mainly aimed at studying the types of substitution used in various socio-political texts. Unfortunately, the authors did not provide a comparison of the accuracy of anaphoric relations detection.

Some researches [6, 9, 10] propose to solve the problem of anaphora detection using machine learning methods. In particular, the authors of the work [9] observed that if the support vector machine (SVM) is used in addition to a set of rules, then the best accuracy is 52.04%. The study [6] found that additional knowledge about the semantic roles of anaphora and antecedent can improve the quality of the solution of the problem by 0.1–6.6%.

The study [2] describes pronominal anaphora resolution in analysis of user opinion data. The authors used 16 characteristics, divided into three categories: anaphoric pronouns, candidates for antecedents and relationship characteristics. A relatively small corpus in the Basque language was taken for training and testing. It consisted of 50 thousand words and 249 anaphoric pronouns. Various methods of machine learning were compared in the experiment: support vector method (SVM), ensemble of decision trees (RF), k-nearest neighbors (kNN) method, multilayer perceptron (MLP), Bayes method (NB), Bayesian combined approach and Decision trees (NB-Tree). Quality assessment was carried out using 10-fold cross-valuation. The results of the experiments showed that a high accuracy of 0.803 is observed for the SVM, while the best recall of 0.702 and the F-measure of 68.3% were obtained using the RF.

An article [7] describes the experiment on the anaphora resolution for the Russian language using a hybrid approach. First, a set of potential antecedents is selected for each pronoun. Next, the most likely candidate is chosen on the basis of a set of characteristics containing information on compatibility of words, statistical, morphological and syntactic characteristics. After that the Random Forest algorithm is used for classification of feature vectors. The highest accuracy of 71% was obtained on the set of all available features.

A hybrid approach to coreference resolution in the English language is presented in [5]. The authors proposed 10 models based on the rules as features for machine learning. For example, the rule “Is there an anaphor-antecedent pair in direct speech?” can be considered as a binary categorical feature. Some of the rules appeared in the article [5] were applied in our work.

The method proposed in our paper is based on machine learning and implementation of a complex approach to feature matrix generation. The feature matrix contains features obtained from the rules or generated from other features. To analyze the syntactic context, a neural network algorithm SyntaxNet [8] was used.

2. Data preparation

We used a text corpus1 of 2,684 texts on criminalistic topics from the informational portal mvd.ru to train our model. The mean text length is about 1,200 symbols. 1,000 more texts were taken for testing. The texts were annotated manually by expert linguists.

Required data were extracted from the texts and transformed into a feature matrix. The matrix rows are represented by all possible pronoun-noun pairs of a single text, some of which are correct anaphoric pairs. The correct pairs got the positive class labels according to the annotation. Thus, the anaphora resolution task was reduced to the binary classification of pronoun-noun pairs.

In the first experiments we searched the antecedents for current pronoun throughout the whole text. But it was founded out that it is rather meaningless and moreover, requires a very large amount of computational resources. The probability of finding an antecedent far away from the pronoun is too low to consider such cases. The experiments described in [9] showed that the optimal window for searching the antecedents is 23 words. Given that the average sentence length in Russian is roughly equal to 10 words, we decided to limit the window to two sentences before and two sentences after the sentence with antecedent. The antecedents that appear after the pronoun are the cases of cataphora. Cataphoric pairs accounted for 33% of all data. The size of training sample was 262,804 pairs pronoun-noun.

Another positive effect of such restriction was partial solution of the imbalanced set problem. The percentage of correct anaphoric pairs was 3% before the limitation, and it increased to 10% after setting a window.

3. Feature Generation and Selection

The feature matrix contains features of anaphor and antecedents which are based on morphological, syntactic, statistical and vector analysis of texts. The whole number of generated features was 2,596, but after the selection only 240 features left.

Feature selection was semi-automatic. The features were ranked by their importance, evaluated by the Extra Trees model, which used for classification. Moreover, the Recursive Feature Elimination (RFE) method was used. The features were divided into several groups, then random features were sequentially removed from each group, and the quality of classification with the current feature group was estimated.

It does not seem possible to describe separately all the used features in the scope of this article because of their large number, but it is possible to combine features into the following groups (see Table 1).

[1] This corpus is available at https://github.com/my-master/CoreferenceData

Table 1. Feature groups of training set

Group description	Feature origin	Number of features
1. *Binary categorical features*, obtained with mystem morphological analyzer and relatively complex rules (for example, "the entity indicated by the antecedent is a person", "anaphor and antecedent agree in person, number and case").	rules, mystem	7
2. *Non-binary categorical features*. They are based on simple grammatical characteristics of anaphor and antecedent (part of speech, number, gender, case, animacy, type of anaphoric pronoun, syntactic relations).	mystem, Syntaxnet	82 (after binarization)
3. *Numerical features* derived from Word2Vec vectors for syntactic contexts. They include all possible distances between context vectors of antecedent and anaphor and distances between antecedent and anaphor own vectors and average vectors of their contexts.	Word2Vec, SyntaxNet	28
4. *Numerical features*, obtained as a result of calculating various linear (i.e., not syntactic) distances, for example, distance in words, sentences, nouns, verbs between anaphor and antecedent.	rules, mystem	13
5. *Transposed vectors*, obtained using SyntaxNet with TF-IDF vectors of morphological and syntactic tags, taken for anaphor and antecedent in three directions of the syntactic context: the child nodes, the parent node, and the sibling nodes.	SyntaxNet	110

4. Classification Process

We considered the problem of anaphora resolution as a binary classification problem of possible pronoun-noun pairs. In view of the large space dimension and the variety of ways to obtain features, it was decided to use the algorithm based on decision trees as the classification algorithm. It was revealed that the Extra Trees [4] algorithm, which is a modification of a random forest, shows the best results. Therefore, it was chosen as the main algorithm.

For the Extra Trees algorithm, the following parameters were selected:

- the maximum percentage of features for finding the best partition was 0.23;
- the number of trees in the forest was 200;
- balanced class weighting method was chosen.

In addition to choosing the main algorithm, it was also necessary to solve the problem with an imbalanced sample, since after the initial screening of incorrect anaphoric pairs by a three-sentence frame, the proportion of correct pairs was still at 10%, which could lead to low accuracy of the trained algorithm. To solve this problem, various simple methods were tested (reducing the number of objects of the major class, duplication of the objects of the minor class), which did not bring a gain in quality. Nevertheless, after applying the ensemble algorithm Balance Cascade F-measure improved by 1%.

The following parameters for the Balance Cascade were selected:

- fraction of the minor class: 0.5;
- maximum number of generated sub-samples: 200;
- random-forest was chosen as internal classifier for quality assessment.

After applying the Balance Cascade algorithm, new true anaphoric pairs were generated and some of the old incorrect pairs were discarded. As a result, the proportion of true pair from the entire sample has already become 25%. However, due to the specificity of the Balance Cascade algorithm, the size of the training sample increased approximately 1.3 times, which increased the requirements for computing power. For example, before the application of Balance Cascade, the size of the training matrix was 262,804 × 240. After converting the sample, the matrix size varies from 341,900 × 240 to 345,800 × 240.

5. Experiment results

Precision, recall and F-measure were used to assess the quality of the proposed method. It is necessary to take into account both precision and recall simultaneously for evaluating the results. Fig. 1 shows the Precision-Recall curve.

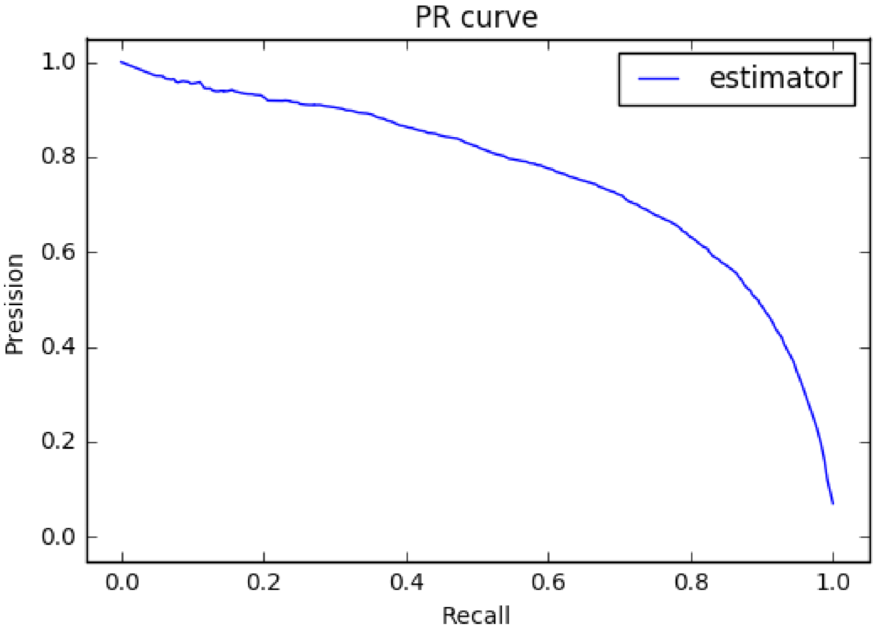


Fig. 1. Precision-Recall curve

Accuracy was not taken into account, since under the conditions of an extremely imbalanced sample it would be high even with a constant classifier that assigns the value of the wrong pair to all pairs. Table 2 gives the best obtained values of precision and recall for a certain threshold of the probability of belonging to the class of correct anaphoric pairs.

Table 2. Precision and recall values obtained in the control sample

Threshold of the probability	Precision	Recall
0.280	0.6577	0.7789
0.285	0.6605	0.7748

Due to high relevance of the feature groups it was decided to test the algorithm quality on each group and their combinations separately. Also it helped to understand the contribution of each group to the overall result. The F-score obtained on different feature groups is shown at the Table 3.

Table 3. F-score on feature groups

Feature group	F-score, %
Binary categorical features	28.6
Non-binary categorical features	57.8
Numerical features derived from Word2Vec vectors	50.0
Numerical features (linear distances)	32.9
Transposed vectors	61.2
Binary categorical features *Non-binary categorical features* *Numerical features derived from Word2Vec vectors* *Numerical features (linear distances)*	70.3
Binary categorical features *Non-binary categorical features* *Numerical features derived from Word2Vec vectors* *Transposed vectors*	70.2
Non-binary categorical features *Numerical features derived from Word2Vec vectors* *Numerical features (linear distances)* *Transposed vectors*	70.1
Binary categorical features *Non-binary categorical features* *Numerical features (linear distances)* *Transposed vectors*	70.9
Binary categorical features *Non-binary categorical features* *Numerical features derived from Word2Vec vectors* *Numerical features (linear distances)* *Transposed vectors*	**71.4**

It can be seen that in the cases when only one of the feature groups was taken into account, the corresponding F-scores differ greatly from each other. The best value of 61.2% is achieved on the transposed vectors obtained using SyntaxNet with TF-IDF vectors. Presumably this is due to the fact that the fifth feature group is the most numerous, i.e. we can see that there is a correlation between the number of features in each group and the obtained result.

At the same time, in the cases when different combinations of feature groups are used simultaneously, their corresponding F-measures differ insignificantly. It implies that despite the correlation among the features, decreasing their number doesn't lead to increase of the F-measure. The best value of 71.4% was obtained in the case when all five feature groups were used.

6. Conclusion

In this article, we offer a complex approach to the anaphora resolution in the Russian language. Formally, the problem of anaphora resolution can be represented as a binary classification problem. The feature matrix for classification contains information about morphological, syntactic, statistical and vector analysis of texts. The total number of generated features was 2,596, but after the selection only 240 the most important features left. All the features can be divided into five groups. Despite the correlation among the features, decreasing their number does not lead to increase of the F-measure.

Acknowledgment

The work was carried out at the Novosibirsk State University with the financial support of the Ministry of Education and Science of the Russian Federation (contract No. 02.G25.31.0146) as part of the implementation of RF Government Decision No. 218 “On State Support Measures for the Development of Cooperation between Russian Higher Educational Institutions and Organizations Implementing Comprehensive Projects for High-Tech Production”.

References

1. *Abramov V., Abramova N., Nekrasova E., Ross G.* (2011), Statistical Analysis of the Coherence of Texts on Social and Political Issues [Statisticheskij analiz svjaznosti tekstov po obshhestvenno-politicheskoj tematike], Proceedings of the 13th All-Russian Scientific Conference “Digital Libraries: Advanced Methods and Technologies, Digital Collections RCDL’2011” [Trudy 13j Vserossijskoj nauchnoj konferencii «Jelektronnye biblioteki: perspektivnye metody i tehnologii, jelektronnye kollekcii» — RCDL’2011], Voronezh, Russia, pp. 127–133.
2. *Arregi O., Ceberio K., Díaz de Illarraza A., Goenaga I., Sierra B., Zelaia A.* (2010), Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque, Procesamiento del language natural, N 45, pp. 291–294.

3. *Barbu C., Mitkov R.* (2001), Evaluation tool for rule-based anaphora resolution methods, Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pp. 34–41.
4. *Geurts P., Ernst D., Wehenkel L.* (2006), Extremely randomized trees, Machine Learning, Vol. 63, N 1, pp. 3–42.
5. *Jurafsky D., Lee H., Chang A., Peirsman Y., Chambers N., Surdeanu M.* (2013), Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules, Association for Computational Linguistics, Vol. 39, N 4, pp. 885–916.
6. *Kamenskaya M. A., Khramoin I. V., Smirnov I. V.* (2014), Data-driven Methods for Anaphora Resolution of Russian, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue" (2014), Issue 13 (20), pp. 241–250.
7. *Malkovskiy M. G., Starostin A. S., Shilov I. A.* (2013), Method of pronoun anaphora resolution in parallel with syntactic analysis [Metod razreshenija mestoimennoj anafory v processe sintaksicheskogo analiza], Collection of scientific works SWorld on materials of the international scientific-practical conference [Sbornik nauchnyh trudov SWorld po materialam mezhdunarodnoj nauchno-prakticheskoj konferencii], Vol. 11, N 4, pp. 41–49.
8. *Petrov S.* (2016), Announcing SyntaxNet: The World's Most Accurate Parser Goes Open Source, available at: https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html
9. *Protopopova E. V., Bodrova A. A., Volskaya S. A., Krylova I. V., Chuchunkov A. S., Alexeeva S. V., Bocharov V. V., Granovsky D. V.* (2014), Anaphoric Annotation and Corpus-Based Anaphora Resolution: An Experiment, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue" (2014), Issue 13 (20), pp. 562–571. URL: http://www.dialog-21.ru/media/1125/dialogue2014_full_version.pdf
10. *Tolpegin P. V.* (2008), Automatic coreference resolution of third person pronouns in Russian texts [Avtomaticheskoe razreshenie koreferencii mestoimenij tret'ego lica russkojazychnyh tekstov], Theses for the degree of candidate of technical sciences, Moscow, 241 p.

PART-OF-SPEECH TAGGING: THE POWER OF THE LINEAR SVM-BASED FILTRATION METHOD FOR RUSSIAN LANGUAGE

Kazennikov A. O. (kazennikov@iqmen.ru)

IQMen LLC, Moscow, Russia

We present our approach to Part-of-Speech tagging and lemmatization tasks for Russian language in the context of MorphoRuEval-2017 Shared Task. The approach ranked second on the closed track and on several test subsets it ranked first.

We proposed a filtration-based method which seamlessly integrates a classical morphological analyzer approach with machine learning based filtering. The method addresses both tasks in a unified fashion. Our method consists of two stages. On the first stage we generate a set of candidate substitutions which simultaneously recovers the normal form and provides all necessary morphological information. We select an optimal substitution for the current word given its context on the second stage.

The filtration stage of the presented method is based on Linear SVMs extended with hash kernel. The extension reduces the size of our model by an order of magnitude and allows to easily tune the tradeoff between the precision and the model size.

Keywords: POS Tagging, Morphological analysis, SVM, Hashing trick

МОРФОЛОГИЧЕСКИЙ АНАЛИЗ: ФИЛЬТРАЦИОННЫЙ МЕТОД НА ОСНОВЕ SVM ДЛЯ РУССКОГО ЯЗЫКА

Казенников А. О. (kazennikov@iqmen.ru)

ЗАО «Айкумен-ИБС», Москва, Россия

В настоящей статье представлен метод снятия морфологической омонимии занявший второе место в общей таблице на закрытой дорожке соревнования MorphoRuEval-2017. Предлагаемый метод сочетает классический морфологический анализ и позволяет одновременно решать задачи лемматизации и восстановления морфологических признаков. Предлагаемый метод состоит из двух стадий: генерации возможных вариантов анализа словоформы и выбора корректной из списка возможных вариантов.

Первая стадия основана на анализе по словарю, состоящего из различных источников: конвертированного словаря AOT, словаря, составленного по корпусу и предиктивного модуля. Вторая стадия реализована с помощью классификации на основе линейной SVM, дополненной алгоритмами хеширования. Это позволяет сократить модель признаков машинного обучения на порядок без какой-либо потери в качестве и в дальнейшем гибко настраивать соотношение между точностью снятия омонимии и размером модели.

Ключевые слова: Морфологический анализ, Снятие морфологической омонимии, SVM, Хеширование

1. Introduction

Morphological analysis plays an important role in almost any NLP pipeline, especially for morphologically-rich languages such as Russian language. It is usually one of the early stages of the pipeline, and the overall performance heavily depends on the quality of these first stages.

There exists a slight ambiguity in the formulation of the part-of-speech tagging problem. Early research on the problem was done mainly for the English language which has a relatively simple morphology, if compared, for example to the Russian language. So the term “part-of-speech” for English usually refers to an extended atomic tagset, rather than a strict part-of-speech tags such as “noun”, “verb”, or “adjective”. The distinction between strict POS tags and the extended atomic tagset is much higher for Russian, which has about 10 strict part-of-speech tags, whereas the full morphological model contains about 10 additional categories witch totals to about 40–60 morphological features (those numbers depend on the used morphological model), and results to over 300 atomic part-of-speech tags. This leads to severe precision penalties when successful approaches for English atomic POS tags are transferred to Russian language without modifications.

The second goal of the Shared Task, the lemmatization, is the task of reconstruction of the normal form of a word and is tightly coupled with the task of POS-tagging. This problem is more significant for the Russian language, because it is the highly inflected language. For example, the Zaliznyak’s dictionary[1] used in AOT project[2] contains about 120k word records which produce on expansion over 4.5M wordforms. That ratio is an order of magnitude higher if compared with English language.

Through this paper we will refer to “POS-tagging” as the task of recovery of a full set of morphological features for a word, and to “morphological analysis” as the joint task of POS-tagging and lemmatization.

The rest of the paper is organized as follows. Section 2 presents related work to the Shared Task. Section 3 describes the MorphoRuEval Shared Task setting. Section 4 introduces our approach to the MorphoRuEval POS-tagging and lemmatization tasks. Section 5 provides the evaluation results. Finally, we provide some concluding remarks in the last Section.

2. Related work

We identify three areas of research related to the MorphoRuEval Shared Task. The first area is the theoretic area of research of the tagset structure that could represent the linguistic properties of the Russian language. In this area we want to note the tagset of AOT project [2], the RusCorpora tagset [3], the SynTagRus tagset [4], the positional tagset [5], and the Universal Dependencies tagset [6].

The second area of research focuses on practical aspects of morphological analysis—the implementation of morphological analyzers. The approach of [7] is based on two separate finite state automata (FSA) for stems and endings, AOT project [2] uses a single automaton for storing the dictionary, the ETAP-3 NLP Processor [8] uses the idea of two-level Finite State Transducer for storing data for both analysis and morphological generation in a single FST. This area includes the research on predictive morphological analysis of unknown words. There we want to note the work of [2] which uses reverse endings to build a guesser FSA to deal with unknown words and [9] that introduces the normalizing substitution concept and presents some heuristics to lexical disambiguation.

The third area related to the Shared Task is the area of POS-tagging and disambiguation. The state-of-the art approaches are based on machine learning techniques. The notable approaches are the transformation-based approach of Brill tagger [10], the decision tree approach of TreeTagger [11], the classical approach based on HMMs of TnT tagger [12] and SVM-based approach of SVMTool [13], further elaborated in [14]. The recent research focuses on deep-learning approaches and various architectures [15, 16, 17].

3. MorphoRuEval Shared Task setting

All participants of the MorphoRuEval Shared Task were provided with several resources to train their models. Some of these resources were annotated and some were plain-text. We will focus on annotated resources only. They included:

1. GICR corpus, 1M tokens.
2. RNC corpus (the open part), 1.2M tokens.
3. SynTagRus corpus, 900k tokens.
4. OpenCorpora corpus, 400k tokens.

All corpora were converted to a simplified variant of Universal Dependencies morphological tagset format [6] (Table 1). The morphological model used in the Shared Task consisted of 12 POS tags and 12 feature categories. A valid parse contains at most one feature from each category. This totals in 40 features (of which 12 were POS tags). All corpora were semi-automatically converted to the Shared Task tagset format. This resulted in some inconsistencies between corpora. However, there were explicitly stated that all inconsistencies should be resolved in the favor of the GICR annotation flavor. Thus, the GICR corpus could be viewed as a gold-standard corpus, and the others as a source of potentially unreliable auxiliary information.

Table 1 sums up the morphological tagset used through the Shared Task. We should note that punctuation marks were treated as words too.

Table 1. Morphological model of the MorphoRuEval-2017. Features skipped from the evaluation are marked with '*'

#	Category	Features
1	POS	NOUN, PROPN (same as NOUN), ADJ, PRON, NUM, VERB, ADV, DET, CONJ*, ADP, PART*, H*, INTJ*, PUNCT
2	Case	Nom, Gen, Dat, Acc, Loc, Ins
3	Number	Sing, Plur
4	Gender	Masc, Fem, Neut
5	Animacy	Anim*, Inan*
6	Tense	Past, Notpast
7	Person	1, 2, 3

#	Category	Features
8	VerbForm	Inf, Fin, Conv
9	Mood	Ind, Imp
10	Variant	Short/Brev
11	Degree	Pos, Cmp
12	NumForm	Digit

Table 2 presents some statistical properties of the provided corpora. It shows significant annotation inconsistencies between corpora used in the Shared Task.

Table 2. Training corpora statistics

Corpus	Tokens	Unique lemmas	Unique feature sets	Unique words
GICR	1M	43k	303	115k
SynTagRus	0.9M	43k	250	104k
RNC	1.2M	53k	557	127k
OpenCorpora	0.4M	42.5k	337	79k

The MorphoRuEval Shared Task had a strong focus on the evaluation of morphological aspects limiting the possible error sources. Both training and testing were done on pre-tokenized data, discarding any errors that could happen due to tokenization differences.

4. Proposed method

Our method integrates a classical dictionary-based morphological analysis pipeline with machine learning based disambiguation techniques.

The overall tagging procedure is straightforward and proceeds in greedy manner. It consists of the following steps:

1. Generate all parse candidates for each token of the sentence
2. Scan the sentence in the left-to-right manner.
    1. Score each parse candidate with respect to the sentence context
    2. Select the best parse
    3. Assign it to the current token
    4. Proceed to the next token

4.1. Candidate generation stage

The first stage of our model generates parse candidates for the given word. We used the normalizing substitution concept from [9] to represent a single parse candidate. A substitution is a triple of:

- The wordform ending
- The Normal form ending
- The full set of associated morphological features, including the POS tag.

This representation simultaneously provides candidate solutions for both goals of the Shared Task: it recovers morphological features of the word as well as the normal form.

A substitution is applied to the word in a trivial manner:

1. The ending is stripped from the word form,
2. The ending of normal form is appended,
3. All morphological features of the substitution are assigned to the parse.

For example, a substitution of

> (wfEnd="e", nfEnd="a", feats=NOUN, Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing)

transforms the word "*руке*" into "*рука*" and assigns respective features to the parse.

We used several data sources to build this module:

- A dictionary collected from the provided corpora, as it is the gold standard for features and lemmatization (after some experiments we used GICR corpus only).
- A partial transformation of the dictionary of AOT project [2] to the Shared Task tagset (the substitution mapping was performed on GICR joined with SynTagRus).
- A guesser for treating unrecognized words (we used GICR only again).
- Some simple heuristics for parsing special kinds of tokens (numbers, for example).
- A hand-crafted dictionary of frequent incorrectly parsed words (~50 wordforms total).

We collected the corpus-based dictionary at the first step. So we got a mapping from each wordform to a set of possible normalizing substitutions.

The conversion of the AOT dictionary posed some challenges. The Shared Task tagset doesn't maps one-to-one to any existing machine-readable dictionary of Russian. We designed a conversion procedure that maps normalizing substitutions of the corpus dictionary to the substitutions of AOT dictionary. We assume that if a corpus substitution perfectly matches an AOT dictionary substitution, then we could safely assign this corpus substitution to other AOT dictionary wordforms that derive this substitution. To do this, we used some transformations of AOT tagset to obtain a partially converted dictionary. Those transformations included:

- Rule-based direct feature mapping. For example "C" → "Noun"
- Splitting verb paradigms to verbs and participles (as they were treated as adjectives through the Shared Task)
- Conversion of short forms of adjectives to adverbs
- Post-processing of the immutable words like *кофе*.

To recover the full mapping we filtered the corpus dictionary through that partially-converted dictionary. We kept only substitution mappings that didn't produce any ambiguities. That led to a conversion of about a third of AOT dictionary substitutions and totaled in 1.6M converted wordforms.

Finally, we implemented a morphological guesser to get viable parses for out-of-dictionary words. The guesser was designed under assumptions of that: a) all irregular words are contained in the dictionary; b) unknown words are relatively long;

c) all unknown words are derived from high-frequency word paradigms. The main idea of the guesser was inspired by [7]. We built two finite-state automata. One for the reversed endings of the word and another one for the reversed stems (prepended with the ending). For example wordform “*руке*” will be split into “е” as ending (id=42) and “кур” as reversed stem. The guess procedure is:

- Reverse the unknown word
- Traverse the endings FSA to find all possible endings
- For each ending, traverse the stems FSA and collect all possible substitutions
- Filter out unreliable parses (for example, if the recognized part is shorter than 3 characters)

At last we added small hand-crafted dictionary of frequent incorrectly-parsed words from other Shared Task corpora as they could appear in the test set. That included some words from Shared Task tagging rules (for example, tagging “*нет*” as a verb), and high frequency adverb/adjective ambiguities missing from AOT dictionary.

4.2. Filtering stage

The filtering stage selects a single parse from a set of generated parse candidates at the previous stage. The overall architecture was inspired by the SVMTool [12] and was further elaborated in [13]. The filtering algorithm is quite trivial: score each parse of the word against the context and choose the highest-scoring one.

The Shared Task tagset contains over 300 different combinations of morphological features. Using a 300-class classifier seemed highly impractical as it doesn’t take advantage of the tagset structure and secondly, that the provided datasets were relatively small and highly imbalanced for this approach.

We trained a separate classifier for each group of features separately. This resulted in 12 multiclass classifiers instead of 300 binary ones.

The following tagging procedure was used:

1. Collect all morphological features from each parse candidate. This step reduces the number of classifier evaluations.
2. Score each feature against the context of the word.
3. Select a parse that:
    1. Has the highest ranking part-of-speech
    2. Has the maximal sum of feature scores.

The selection procedure was split into two parts to prevent the case when the sum of feature scores outweights the part-of-speech score. It is undesirable as we found out that part-of-speech classifier has a negligible error rate (about 0,8%).

4.3. Feature group classifier and the context feature model

We used a modified SVM multiclass classifier of LIBLINEAR [17] to score a single feature group. It uses one-against-all classification scheme and the training algorithm optimizes all classes simultaneously. We modified the original implementation by replacing a weight vector for each class by a shared vector by means of the hashing trick [18]. The basic idea is the replacement of the dot-product function:

$$dot(w, x, i) = \sum_j w[i][j]x[j]$$

by:

$$dot(w, x, i) = \sum_j w[hash(i, j)]x[j]$$

where w is a weight vector, x is a feature vector, i is the class we are scoring against, and $hash(i, j)$ is a hash function that maps its inputs to an integer value from a predefined range (regarded as effective feature count). Our system used MurmurHash3[19] as the hashing function and 2M as effective feature count. The effective feature count is independent of the number of classes, so the per-class effective feature count is a fraction of the total feature space. For example, if the effective feature count is 1M and the number of classes is 10 then the effective feature count per class is just 100k.

The hashing trick allows to easily tune the resulting feature space size. Another benefit of the hashing trick is that we discarded the feature mapping table of the one-hot encoding procedure and significantly reduced memory requirements for our method.

The drawback of the hashing trick is in its lossy compression scheme. And if the chosen effective feature count is too small for the problem, the hash function collisions could significantly reduce the model performance.

We estimated the total number of distinct features of our model and used it as an initial effective feature count. We tried to double the number of effective features and haven't seen any significant performance improvement. After that, we tried to halve the effective feature count and observed some performance loss. So we used the original estimation of the effective feature count through all experiments.

Our feature model produces about 3M distinct features. The hashing technique reduced the effective per-class feature count by an order of magnitude without significant performance loss.

The model uses mostly context features. We used a context window of size 7 (±3 words around of the main one). The context window was divided into two parts: the tagged part (words before the current one), and untagged one (words starting with the current one). All parses in the tagged part were already resolved and we could use all available information (such as case, number gender features) from them.

The features used for the tagged part of the context were inapplicable because words in the untagged part don't have a resolved parse yet. To overcome this we used a concept of *ambiguity class* over the morphological category. It is a sorted set of the possible morphological features of that category collected from candidate parses of the word. For example, for wordform "человека" the ambiguity class over the "Case" category would be "genitive/accusative", because we don't know the correct case for the word yet, but we can narrow it to two options instead of six.

For each word of the full context we use following features:

- word prefixes of length 2, 3 and 4
- word suffixes of length 2, 3 and 4
- wordform itself
- lowercased wordform
- For each word of the tagged part of the context:

- POS tag of the word
- POS tag + suffix
- POS tag + suffix of the main word
- Number, Gender, Case morphological categories of the word (and their combinations)
- Stem and the Ending

For each word in untagged part of the context (starting from the main one):

- Ambiguity classes for POS, Number, Gender and Case categories
- Ambiguity classes for POS, Number, Gender and Case categories coupled with suffix of the main word

Finally, for the main word we used some additional features:

- A flag for the main word is at start of the sentence
- Capitalization of the main word

5. Experiments

We conducted several experiments on the different combinations of training/test data during the development of our system. The results are presented in Table 3.

Table 3. Evaluation of our model on different training/test set combinations

Training/Test pair	POS-only,	POS-full	Lemma	Lemma+POS
GICR/GICR (9:1)	99,23%	94,52%	98,59%	76,28%
Syntagrus/Syntagrus (9:1)	97,85%	91,78%	97,73%	58,22%
RNC/RNC (9:1)	96,64%	70,28%	94,08%	25,33%
OpenCorpora/OpenCorpora (9:1)	98,17%	57,29%	98,51%	14,53%
GICR/Syntagrus	96,24%	88,85%	97,26%	48,81%
GICR/RNC	95,18%	68,64%	93,67%	23,77%
GICR/Opencorpora	97,11%	55,91%	97,93%	13,61%

Table 3 shows a significant loss of precision when the model was trained on one corpus and tested on a different one. The Shared Task organizers explicitly stated that the GICR annotation could be viewed as a reference and all inconsistencies should be resolved in favor of GICR annotation. As a result, we tuned our model to the GICR annotation.

Table 4. Effect of using partially-converted AOT dictionary, GICR corpus

Training/Test pair	POS-only,	POS-full	Lemma	Lemma+POS
Guesser only	98.22%	91.99%	86.02%	45.45%
Guesser + Corpus Dict	99.04%	94.37%	98.96%	76.67%
Guesser + Corpus dict + AOT Dict	99.23%	94.52%	98.59%	76.28%

We note high sensitivity of the proposed model to the quality of the generation stage (Table 4). The “guesser only” mode generates all parse candidates guesser only. The “guesser + corpus dict” experiment show synthetic results when the parse candidates of the GICR corpus dictionary were complemented heuristically by the guesser results (to handle the situation when there is a potential parse of the word that didn't occur in the corpus). Our final model (Guesser + Corpus dict + AOT Dict in the table) shows significant improvement from the proposed corpus-dictionary mapping procedure.

Our final model was trained on the GICR corpus. Our final results on the closed track of the Shared Task are presented in Tables 5–8. Our results are marked with bold, the best ones are marked by '*'.

Table 5. Precision on News subset of the test set

Team ID	Tags, by word	Tags, by sentence	Lemmas, by word	Lemmas, by sentence
O	**93.99%***	**63.13%**	**92.96%**	**54.62%***
A	93.83%	61.45%	93.01%*	54.19%
C	93.71%	64.8%*	—	—
H	93.35%	55.03%	81.6%	17.04%

Table 6. Precision on Vkontakte subset of the test set

Team	Tags, by word	Tags, by sentence	Lemmas, by word	Lemmas, by sentence
H	92.42%*	63.59%	82.8%	35.39%
O	**92.39%**	**64.08%**	**91.69%***	**61.09%***
C	92.29%	65.85%*	—	—
A	91.49%	61.44%	90.97%	60.21%

Table 7. Precision on Fiction subset of the test set

Team	Tags, by word	Tags, by sentence	Lemmas, by word	Lemmas, by sentence
C	94.16%*	65.23%*	—	—
O	**92.87%**	**60.91%**	**92.01%***	**57.11%***
A	92.4%	60.15%	91.46%	55.08%
H	92.16%	56.6%	77.78%	22.08%

Table 8. Precision on full test set

Team	Tags, by word	Tags, by sentence	Lemmas, by word	Lemmas, by sentence
C	93.39%*	65.29%*	—	—
O	**93.08%**	**62.71%**	**92.22%***	**58.21%***
H	92.64%	58.4%	80.71%	25.01%
A	92.57%	61.01%	91.98%	56.49%

Our approach ranked second, losing to the top system slightly more than 0.3% on POS-tagging task. On News subset our system showed top precision. On the

Vkontakte subset our system lost about 0.04% to the top one. The result tables show that our method is strongly consistent and robust across different text sources types.

On the lemmatization task, our approach ranked top, seconding only in the News subset with the gap of only 0,05%. The lemmatization performance was also consistent across different text sources types.

6. Conclusions

We presented an approach to the part-of-speech tagging and lemmatization that is closely related to classical morphological analysis frameworks. The two-stage scheme showed high precision and robustness. That allowed our model to get a consistent second rank on the POS-tagging task of the closed track of the MorphoRuEval-2017 Shared Task, even ranking first on several test subsets. Our method ranked first on the full test set of the lemmatization task, ranking second only on News subset with the gap of 0,05% to the top system.

Experiments showed that the model performance significantly depends on the consistency of the corpus annotation and for this level of precision corpora-to-corpora differences are critical to the model performance.

The application of the converted AOT dictionary significantly improved the overall performance of our method. The consistency of morphological information between the generation phase of our model and gold standard corpus also was critical to the success of our method.

We believe that the performance of the presented method could be improved by further efforts on dictionary-to-corpus matching.

The source code for all experiments is available at: https://github.com/kzn/morphoRuEval.

Acknowledgements

We want to thank MorphoRuEval organizers team for their work in organizing this Shared Task competition.

References

1. *Zaliznyak, A. A.* (1980). Russian grammatical dictionary [Grammaticheskyi slovar' russkogo yazyka]. Russkij Jazyk, Moskva.
2. *Sokirko A. V.* (2004) Morphological Modules on AOT.RU [Morfologicheskie moduli na saite AOT.RU]. Dialogue conference proc.Moscow, pp. 559–564.
3. *Lyashevskaya O. N., Plungian V. A., Sichinava D. V.* (2005) On the morphological standard for Russian National Corpus for Russian Language [O morphologicheskom standarte Natsional'nogo korpusa russkogo yazyka]. Natsional'nyi Korpus Russkogo Yazyka, 2005(2), pp. 111–135.
4. *Apresyan Yu. D., Boguslavsky I. M., Iomdin B. L., Iomdin L. L., et al.* (2005). Syntatic and semantic annotated corpus for Russian language: state-of-the-art and perspectives [Sintaksichesky i semantichesky annotirovannyi korpus russkogo

yazyka: sovremennoe sostoyanie i perspektivy]. Natsyonalny corpus russkogo yazyka. 2005, pp. 193–214.

5. *Hana, Jirka, and Anna Feldman* (2010). "A Positional Tagset for Russian." In LREC.
6. *Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajic J., Manning Ch. D., McDonald R., Petrov S., Pyysalo S., Silveira N., Tsarfaty R., Zeman D.* (2016), Universal Dependencies v1: A Multilingual Treebank Collection, Proc. of LREC 2016, Portoroz, Slovenia, pp. 1659–1666.
7. *Segalovich, I.,* (2003). A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. In MLMTA (pp. 273–280).
8. *Kazennikov A.O* (2008). Using Finite Automata for Morphological Analysis and Synthesis Based on the Dictionaries of the ETAP-3 System, [Ispol'zovanie konechnyi avtomatov dlya morphologicheskogo analiza i sinteza na osnove slovarei sistemy ETAP]. Sb. tr. konf. molodykh uchenykh i spetsialistov ITIS (Proc. Conf. Young Scientists and Specialists of ITIS), pp. 201–205
9. *Zelenkov Yu., Segalovich I., Titov V.* (2005) Probabilistic Model for Morphological Disambiguation based on Normalizing Substitutions and Nearest Word Positions [Veroyatnostnaya model' snyatya morfologicheskoy omonimii na osnove normalizuyuyschikh podstanovok i pozitsiy sosednikh slov]. Dialogue conference proc., Moscow pp. 188–197.
10. *Brill, E.* (1992, February) A simple rule-based part of speech tagger. In Proceedings of the workshop on Speech and Natural Language (pp. 112–116). Association for Computational Linguistics.
11. *Schmid, H.* (1995) Treetagger| a language independent part-of-speech tagger. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, 43, p.28.
12. *Brants, T.* (2000) TnT—A Statistical Part-of-Speech Tagger. "6th Applied Natural Language Processing Conference".
13. *Gimenez, J. and Marquez, L.,* (2004) SVMTool: A General POS Tagger Generator Based on Support Vector Machines, Proc. 4 Int. Conf. Language Resourc. Evaluat. (LREC'04), Lisbon, Portugal, pp. 43–46.
14. *Petrochenkov V. V., Kazennikov A. O.* (2013) A Statistical Tagger for Morphological Tagging of Russian Language Texts. Automation and Remote Control, Vol. 74, No. 10, pp. 1724–1732
15. *Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.* (2011) Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), pp. 2493–2537
16. *Huang, Z., Xu, W. and Yu, K.* (2015) Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.
17. *Plank, B., Søgaard, A. and Goldberg, Y.* (2016) Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. arXiv preprint arXiv:1604.05529.
18. *Fan, R.-E., Chang, K.-W., Hsieh, C.-J., et al.* (2008) LIBLINEAR: A Library for Large Linear Classification, J. Machine Learning Res. vol. 9, pp. 1871–1874.
19. *Shi, Q., Petterson, J., Dror, G., et al.* (2009) Hash Kernels for Structured Data, J. Machine Learning, vol. 10, pp. 2615–2637.
20. *Appleby A.* (2008) Murmurhash 3.0. https://github.com/aappleby/smhasher.

ARBITRARINESS OF LINGUISTIC SIGN QUESTIONED: CORRELATION BETWEEN WORD FORM AND MEANING IN RUSSIAN

Kutuzov A. B. (andreku@ifi.uio.no)

University of Oslo, Norway

In this paper, we present the results of preliminary experiments on finding the link between the surface forms of Russian nouns (as represented by their graphic forms) and their meanings (as represented by vectors in a distributional model trained on the Russian National Corpus). We show that there is a strongly significant correlation between these two sides of a linguistic sign (in our case, word). This correlation coefficient is equal to 0.03 as calculated on a set of 1,729 mono-syllabic nouns, and in some subsets of words starting with particular two-letter sequences the correlation raises as high as 0.57. The overall correlation value is higher than the one reported in similar experiments for English (0.016).

Additionally, we report correlation values for the noun subsets related to different phonaesthemes, supposedly represented by the initial characters of these nouns.

Keywords: distributional semantics, word2vec, semantic similarity, edit distance, vector space models, phonosemantics

ПРОИЗВОЛЬНОСТЬ ЯЗЫКОВОГО ЗНАКА ПОД ВОПРОСОМ: КОРРЕЛЯЦИЯ МЕЖДУ ГРАФИЧЕСКОЙ ФОРМОЙ СЛОВ И ИХ ЗНАЧЕНИЕМ В РУССКОМ ЯЗЫКЕ

Кутузов А. В. (andreku@ifi.uio.no)

Университет Осло, Норвегия

Ключевые слова: дистрибутивная семантика, word2vec, семантическая близость, векторные репрезентации лексики, фоносемантика, расстояние Левенштейна

1. Introduction

The arbitrariness of linguistic sign is one the foundational principles in the studies of language since [De Saussure 1916]. It assumes that there is no relationship between the word forms (phonetic or graphematic) and their meanings: any meaning can theoretically be conveyed by any sequence of sounds or characters, and they are mutually independent. This assumption is important for many linguistic problems, and for understanding language as a system in general.

However, there are well-known exceptions from this law. Many languages feature clusters of words with similar meaning, in which some part of their surface form (for example, initial sounds) is consistently reproduced. These reproduced patterns are called *phonaesthemes* [Firth and Strevens 1930] and seem to violate the principle of sign arbitrariness. Examples of phonaesthemes in English include initial sequence "*gl-*" related to vision or light [Bergen 2004]; in Russian one can note the sequence "*-cmp-*" related to quickness or streaming [Mikhalev 2008], etc.

Another exception is *onomatopoeia*: cases when phonetic form of a word is motivated by the actual sound related to the denoted notion (Russian "*мяукать*" *to meow*). In this case, the linguistic sign becomes to some extent iconic and one observes the emergence of clear relationship between the form and the meaning.

It seems obvious that this iconicity can be manifested both in localized phonosemantic sets (groups of words with similar meanings and surface forms) and in the vocabulary of language as a whole (systematicity). It seems interesting to attempt testing the actual robustness of the arbitrariness principle and to measure the degree of systematic iconicity in different languages.

This can be done by measuring the correlation between the semantic and "surface" differences between word pairs. While surface differences can be easily represented with the so-called edit distances, semantic representations of words are more difficult to obtain. However, recent achievements in distributional semantics (manifested by the advent of prediction-based and other machine learning approaches to producing vector representations of words) provided computational linguists with efficient and robust lexical meaning models which can be trained on very large corpora. These models, including the so called *neural embeddings*, exhibit substantial performance in various natural language processing tasks, including prediction of the pairwise similarities between words [Baroni 2014].

In the presented pilot study I attempt to employ neural embeddings to measure the degree of systematic iconicity in the Russian language. I describe a series of experiments with correlations of semantic and orthographic distances between frequent Russian nouns. The results seem to support the hypothesis that there is a statistically significant systematicity in the Russian language, expressed even stronger than that reported for English in [Monaghan et al. 2014].

The paper is organized as follows. In Section 2 I briefly put the research in the context of the previous work. In Section 3, the experimental design is described, together with the data sources I used. Section 4 presents the results of experiments on several datasets and discusses them. In Section 5 I conclude and outline the possible future work.

2. Related work

For English, the initial statistically rigorous experiments in phonosemantic systematicity are described in [Shillcock et al. 2001] and [Monaghan et al. 2014]. They used the Levenshtein distance [Levenshtein 1966] between orthographic word forms and the semantic distances produced by various distributional vector space models, in order to test whether differences in form are accompanied by differences in meaning. Their findings confirmed that there is a statistically significant (though low) correlation between semantic and orthographic distances in the set of mono-morphemic monosyllabic English words. Thus, the form space and the meaning space seem to be related.

Moving to more recent works, I was strongly inspired by the research of [Gutiérrez et al. 2016] in which it was proven that word embedding models can be helpful in studying violations of the arbitrariness principle in English. They also developed a new kernel-based algorithm for learning weights for different operations in the Levenshtein algorithm, which allowed finding local clusters of phonosemantic systematicity with the higher accuracy.

A different vein of research in this direction (not employing distributional semantic models) is represented by [Blasi et al. 2016]. They used Swadesh lexicons for several thousand world languages to trace bias in the frequency with which words denoting certain concepts tend to carry specific phonemes in contrast to their baseline occurrence in other words. They came to the conclusion that strongly expressed sound-meaning associations indeed exist even cross-linguistically.

For Russian, experiments related to systematic iconicity were performed by Alexander Zhuravlev (see, for example, [Zhuravlev 1991]). However, at that time it was impossible to employ large-scale distributional models, and thus opinions of limited number of informants were used to quantify semantic properties of words, rendering the results unstable and difficult to verify. I am not aware of any publications studying correlation between word embedding based semantic distances and graphematic distances for Russian.

Distributional semantic models are essentially based on the assumptions that word meaning is strongly related to the word's typical contexts [Firth 1957]. The meaning of words is represented with the so called *word embeddings*: dense real-valued vectors derived from word co-occurrences in large text corpora. They can be of use in almost any linguistic task related to semantics, and have recently become a buzzword in natural language processing (especially those trained using shallow neural networks). Their increased popularity is mostly due to new prediction-based approaches, which allowed to train distributional models with large amount of raw linguistic data very fast.

Some of the most popular word embedding algorithms in the field are highly efficient *Continuous Skip-Gram* and *Continuous Bag-of-Words*, implemented in the well-known *word2vec* tool. For more details, I refer the reader to [Turney and Pantel 2010] and [Mikolov et al. 2013]; application of these models to Russian is described, among others, in [Kutuzov and Andreev 2015].

3. Experimental setting

3.1. Data sources

I employed Russian National Corpus[1] [Plungian 2005] as the primary source of Russian texts. I also limited myself to nouns in this particular research, leaving other parts of speech to future work.

In order to test systematicity, a set of test words is needed. Ideally, it should consist of mono-morphemic words to exclude the influence of affixal word formation: in the case of "*отдел*" *department* and "*раздел*" *section*, phonetically similar words are generated to denote similar concepts, following straightforward and transparent derivation rules, not some arbitrary connection between the sound/graphical form and the meaning. It means words with shared roots should be avoided in this task.

Automatic morphemic analysis of Russian words is a difficult problem in itself [Lyashevskaya et al. 2009], so for this pilot experiments I assumed that the set of monosyllabic nouns can serve as a sort of proxy to the set of mono-morphemic nouns. I defined a "monosyllabic" word as containing one and only one vowel, and with this in mind, compiled 4 sets of nouns:

1. **Mono**: all monosyllabic nouns with frequency 100 and more in the RNC (1 729 words in total);
2. **Bi**: monosyllabic and bisyllabic words with frequency 1,000 and more in the RNC (2,900 words in total);
3. **Bi_NoDim**: the same as the previous one but excluding the nouns ending with the diminutive suffixes "*-ок*", "*-ек*" and "*-ка*" (2,633 words in total);
4. **All**: all nouns with frequency 1,000 and more in the RNC (6,715 words in total).

In all the datasets, I excluded very short words (less than three characters) and the words containing non-Cyrillic characters and digits. I also filtered out proper names and toponyms as detected by *Mystem* [Segalovich 2003].

The different choice of frequency thresholds is explained by the fact that I strive to achieve two contradictory aims: on the one hand, I need as many words in each dataset as possible (for the detected correlations to be statistically significant), and on the other hand it is desirable for the words to be as frequent as possible, in order for their distributional vectors (embeddings) to be well-trained. The chosen thresholds were selected as a good trade-off, resulting in datasets in the order of several thousand words, similar to the ones used in [Gutiérrez et al. 2016] and other related studies for English.

The main object of our experiments is the **Mono** dataset, as it is supposed to be least influenced by word formation (most words in it do not share roots) and thus its systematicity should best reflect the real relationship between form and meaning in Russian. The other three datasets were compiled for reference and to test what is the amount of influence of word-formation patterns on the phonosemantic systematicity.

[1] Further RNC.

3.2. Distributional model

For computing orthographic distances between words, I used the well-known Levenshtein (edit) distance algorithm [Levenshtein 1966] implemented in *Python*. However, to be able to calculate semantic distances a more sophisticated approach is needed.

To this end, I used the *Continuous Skipgram* distributional algorithm [Mikolov et al. 2013] which learns vectorial representations for words (neural embeddings) based on their co-occurrences in the training corpus. I trained the model on all the RNC texts, using vector size 300 and symmetric context window of 10 words to the left and 10 words to the right, leaving other hyperparameters at their default values. Prior to training, the corpus was tokenized, split into sentences, lemmatized and PoS-tagged using *Mystem*. For training itself, I employed the *Continuous Skipgram* implementation in the *Gensim* library [Řehůřek and Sojka 2010].

Distributional models can be intrinsically tested for their sanity, for example, using semantic similarity or analogy test sets. For the former, I employed the Russian part of *Multilingual SimLex999* [Leviant and Reichart 2016] which contains human judgments on the relative semantic similarity of word pairs, and the task for the model is to mimic the rankings produced by humans. This test set is known to be difficult for distributional models: its authors managed to achieve Spearman rank correlation only as high as 0.26 for Russian with the model they trained on Wikipedia. At the same time, the model used in this research showed a higher correlation of **0.36**.

The analogy test sets pose models with the task to guess one word in a “semantic proportion” (for example, “Rome is related to Italy as Moscow is related to ???”). On the translated Russian version of the *Google Analogies* dataset [Mikolov et al. 2013] the employed model showed accuracy **0.65** (using only semantic sections of the data set). There are no known published results with this translated test set for other Russian models, but the value is comparable to state-of-the-art results for English[2].

Both results also fit well into the average performance of the Russian models featured at the *RusVectores* web service [Kutuzov and Kuzmenko 2017]. Thus, I presuppose that the trained model is good enough and in general outputs correct predictions on the semantic similarities and dissimilarities of Russian words (at least comparable to state-of-the-art).

3.3. Measuring correlation

In order to measure the degree of dependency between the form and the meaning, I first calculated pairwise orthographic (string) and semantic distances between all words in the datasets. The orthographic distance was calculated as Levenshtein edit distance, while the semantic distance was equal to $1 - CosSim$, where $CosSim$ is the cosine similarity between word embeddings in the vector space of the model trained on the RNC. In the rare cases when cosine similarity was negative (about 1.5% of all the pairwise similarities), I assigned it zero value, so as the range of the

[2] See http://www.aclweb.org/aclwiki/index.php?title=Google_analogy_test_set_(State_of_the_art)

cosine distance was within [0...1]. The number of pairwise distances for the dataset of n words is equal to $n\,(n-1)/2$, so I got two sets (edit and cosine) of 1,493,856 distances for the **Mono** datasets, with this number being about 3.5 million for the **Bi_No-Dim** dataset, more than 4 million for the **Bi** dataset, and about 22.5 million for the **All** dataset.

Then, it is trivial to calculate any suitable correlation coefficient between the edit distances and cosine distances, that is, to what extent it is true that one of the parameters grows with the growing of another. Linguistically speaking, high correlation would mean that word pairs similar in form tend to be similar in meaning, and vice versa. Zero correlation would mean that the form and the meaning are absolutely unrelated. As the sets are quite large, I expected the calculated coefficients to be statistically significant, which proved to be true (see below).

The ordinary correlation coefficients are however not enough: they presuppose that the values in the data sets under analysis are independent, and this is not the case for the pairwise distances (changing the representation of one word influences several distances, not only one). Thus, I followed the previous work in testing the significance of the correlation using Mantel permutation test [Mantel 1967].

Mantel test essentially performs random shuffling of the value assignments in one of the two sets (for example, in the semantic distances). It generates a predefined amount of such "possible lexicons" (randomly drawn from the space of all possible permutations), and then computes the ordinary correlation coefficients between orthographic and semantic distances in these generated "lexicons", as well as the correlation for the real lexicon. Then, the proportion of the lexicons that produced higher correlations than the real one is calculated; based on this, the veridical (true) correlation in the real lexicon is found, together with the significance measures. The idea behind this approach is that if the correlation is not accidental, one will very rarely find a higher correlation in randomly generated lexicons.

The most popular correlation measure in the literature is Pearson correlation coefficient. However, there are two reasons against using it with my data:

1. The distances in the sets are not distributed normally: for example, for the semantic distances in the **All** dataset, the normality statistics [D'Agostino and Pearson 1973] is equal to 4,775,600, with $p = 0$ (zero probability that this data can come from a normal distribution).
2. The distances are strongly skewed to the right (see the Figure 1): this is arguably related to the well-known problem of *hubness* in vectorial spaces [Dinu et al. 2014].

Pearson coefficient is known to become non-robust when the data is not normally distributed and particularly when it is skewed. Thus, with my Mantel tests I employed Spearman rank-order correlation coefficient. In fact, for the experiments below, Pearson returned the same results, but I still report Spearman to be on the safe side.

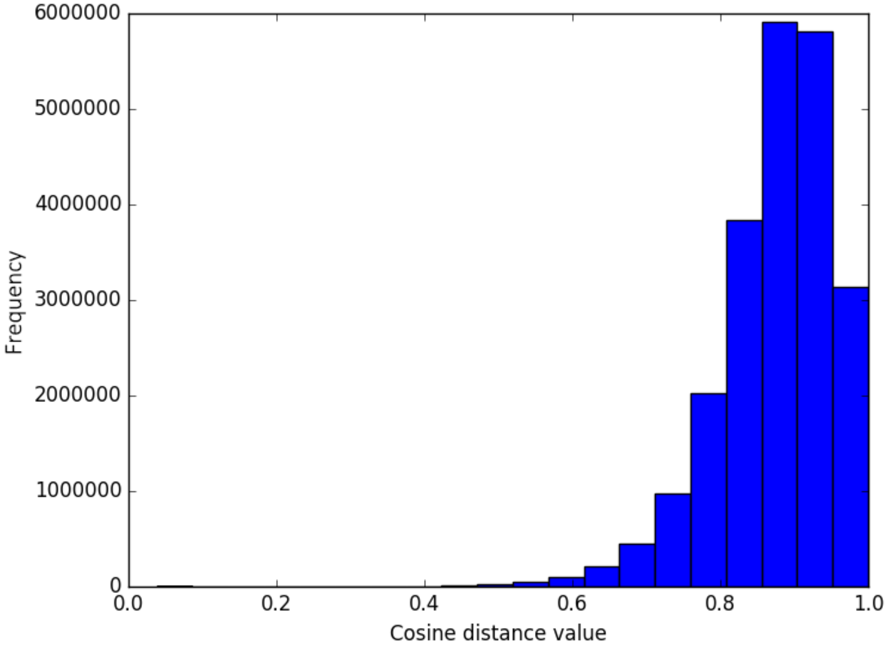


Fig. 1. Distribution of the values of pairwise cosine distances in the **All** dataset

In the next section, I describe the results of the experiments.

4. Results and discussion

I calculated Spearman correlation for all the datasets, using Mantel test with 1,000 random permutations. The results are presented in the Table 1.

As one can see, for the set of monosyllabic words, the correlation between the semantic distances and the orthographic edit distances is about 0.03, with the correlations for the less restricted datasets expectedly higher, reaching 0.08 in the case of all nouns. The value of the correlation coefficient itself is not high, but the Mantel test shows that it is strongly significant: $p = 0.001$ here means that only one lexicon from 1,000 tested has produced the correlation equal or higher to the real one[3]. Of course, this was precisely the real lexicon. Thus, all the random lexicons showed lower correlations, and it is extremely unlikely that the link between edit distances and semantic distances in the real lexicon is accidental.

[3] 10,000 permutations showed exactly the same results ($p = 0.0001$).

Table 1. Correlations between orthographic edit distances and semantic distances

Dataset	Spearman correlation	Mantel test upper-tail *p*-value
Mono	0.0310	0.001
Bi_NoDim	0.0519	0.001
Bi	0.0586	0.001
All	0.0800	0.001

The interesting fact is that in a similar experiment, [Monaghan et al. 2014] reported the correlation of only 0.016 for the set of English mono-morphemic words. The results of the experiments seem to suggest that Russian possesses at least as strong systematicity as English, and probably even stronger. This of course does not disprove the principle of the arbitrariness of linguistic sign in general; however, it is clear that there are some regular exemptions from this law, manifested throughout the lexicon.

Still, the correlation coefficient of 0.03 (and even 0.08) seems to be rather low. Considering that it is statistically very significant, the reason for this might be that the correlation is at least partly "localized" in some parts of the lexicon, not uniformly "dispersed" across all lexemes. In other words, for some nouns the connection between their form and their meaning is stronger than for the others.

One can attempt to trace this local systematicity by segmenting the original dataset into several subsets and measuring correlation for each of them. I performed this experiment on the initial two-character sequences in the **Mono** dataset, splitting it into 321 subsets corresponding to these sequences (for instance, a subset of nouns starting with "*ст-*", etc). Then, I filtered out 159 subsets containing less than three nouns, and 18 subsets with no variance in the pairwise edit distances (for example, the "*чи-*" subset containing the words "*чиж*", "*чик*", "*чин*", "*чип*", "*чиф*", and "*чих*", with all the pairwise edit distances equal to one, leaving no possibility to calculate correlation). This left me with 144 "valid" subsets.

Correlation coefficients were calculated for each of these datasets in the way described above. The distribution of correlation coefficients for all the subsets is shown on the Figure 2 (blue histogram). For some subsets the correlation was almost perfect (close to 1 or -1), but in most cases it was not statistically significant. One example of this phenomenon is the "*тв-*" subset (*"тварь" beast, "твердь" ground, "твист" twist*) with the veridical correlation equal to 1, and the *p*-value equal to 0.17, far above the 0.05 threshold of significance.

Note that it is difficult to mine anything useful from the negative correlation coefficients in this case. First, only 3 of them were statistically significant at 0.05 level. Second, even conceptually, negative correlation here means that words in the subset tend to become more similar in their meaning as the differences in their graphical form grow. This hardly makes any sense, thus I am inclined to consider the negative correlations a statistical fluctuation.

In general, it seems that grouping words by their initial characters indeed reveals local areas of high systematicity. To prove that it is not a statistical illusion, I sampled the **Mono** dataset to produce a comparable collection of 144 random subsets containing

12 words each (without replicating words across subsets), and measured correlations within these subsets. As one can see on the Figure 2 (red histogram), the distribution of correlations in these subsets is much narrower and more normal than in the initial letters based subsets. Correlation values are mostly concentrated around zero (as expected for random data), and what is important, we do not observe subsets with correlations higher than 0.4...0.5, and even those are rare. In contrast, the initial letters based subsets clearly feature many strongly correlated cases, breaking the normal distribution of correlations. This supports the point that the strength of connection between the form and the meaning of words is at least partly conditioned by their initial characters/phonemes.

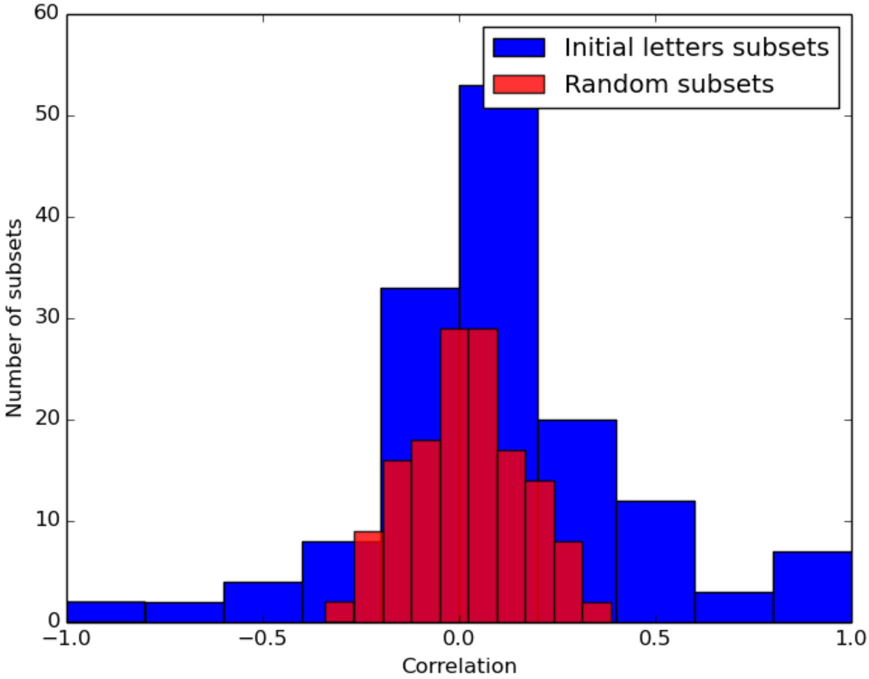


Fig. 2: Distribution of correlation coefficients in the subsets of the **Mono** dataset

The Table 2 presents 10 initial letters based subsets with the highest positive correlation among those which feature p-value less than 0.05 (as I was interested in the cases with the robust signal).

Some found subsets are quite interesting even with simple eyeballing. For instance, the highest correlation is demonstrated by the “*ха-*” subset featuring words like “*хай*” *loud speaking*, “*хам*” *mucker*, and “*харч*” *foodstuff* with this phonaestheme probably related to negative or derogatory connotations (but also “*хаббл*” *Hubble*, “*хадж*” *Hajj*, “*хан*” *khan*, “*хант*” *Khanty*, “*хань*” *Han*, “*хаш*” *khash*). The “*ше-*” subset contains “*шелк*” *silk* and “*шерсть*” *wool* (but also “*шейх*” *sheikh*, “*шельф*” *continental* *shelf*, “*шен*” as a surname, “*шень*” as a surname, “*шер*” as a proper name, “*шест*” *pole*, “*шеф*” *chief*), while in the “*гл-*” subset[4] one sees the nouns “*глубь*” *depth*, “*глушь*” *wilderness* and “*гладь*” *smooth surface*, all associated with natural substances and spaces (but also “*главк*” *department*, “*глад*” *hunger*, “*глаз*” *eye*, “*глас*” *voice*, “*глист*” *helminth*). At the same time, other subsets (like “*дж-*”) seem to be not more than simple clusters of borrowed words: “*джей*” *Jay*, “*джим*” *Jim*, “*джин*” *Gin*, etc.

For certain, most (if not all) of these correlations can be explained with rigorous diachronic research: for example, some words in the pairs can be cognates. However, I still believe that these “pockets of sound symbolism” [Gutiérrez et al. 2016] deserve a closer look[5]. Whatever are the reasons for the statistically significant co-variation of the graphic form and semantics of Russian nouns, it is obvious that this co-variation exists in the present state of the language and it can be quantified. What follows is that the linguistic sign is not as arbitrary as we were used to thinking.

Table 2. Most systematic initial phonaesthemes in the **Mono** dataset

Initial	Correlation	*P*-value	Number of words in the subset
ха-	**0.57**	0.011	9
дж-	0.43	0.047	7
ше-	0.39	0.015	9
фо-	0.35	0.019	9
ва-	0.33	0.017	10
ло-	0.32	0.011	13
ле-	0.27	0.012	14
ка-	0.26	0.029	16
ку-	0.25	0.012	17
ба-	0.22	**0.005**	23

5. Conclusions and future work

I presented the results of preliminary experiments on finding the link between the surface forms of Russian nouns (as represented by their graphic forms) and their meanings (as represented by vectors in a distributional model trained on the Russian National Corpus). I showed that there is a strongly significant correlation between these two sides of word as a linguistic sign. This correlation coefficient is equal to 0.03 as calculated on a set of 1,729 mono-syllabic nouns.

In many subsets of words starting with particular two-character sequences, the correlation (statistically significant) raises as high as 0.3 and more, with one case of 0.57. The overall correlation value is higher than the one reported in similar experiments for English (0.016).

[4] Its *p*-value is 0.055, only slightly exceeding the threshold needed to get to the Table 2.

[5] All the raw data used in this paper is available at http://ltr.uio.no/~andreku/arbitrariness/.

In the future, I plan to refine the datasets by more accurate filtering of noise entities (first of all, abbreviations and proper names) and probably extract mono-morphemic words from one of the available Russian morphemic dictionaries ([Kuznetsova and Efremova 1986], [Tikhonov 2003]). I am also going to enrich the experiments to include other parts of speech except nouns.

Finally, it seems fruitful to employ string metric learning for kernel regression [Gutiérrez et al. 2016] to learn weights for different types of operations in edit distances and thus improve the sensitivity of the Levenshtein metric.

References

1. *Baroni M., Dinu G., Kruszewski, G.* (2014), Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 238–247.
2. *Bergen, B. K.* (2004), The psychological reality of phonaesthemes. Language, 290–311.
3. *Blasi, D. E., Wichmann, S., Hammarström, H., Stadler, P. F., Christiansen, M. H.* (2016), Sound–meaning association biases evidenced across thousands of languages. Proceedings of the National Academy of Sciences, 2016 113 (39) 10818–10823.
4. *D'Agostino R., Pearson E. S.* (1973), Tests for departure from normality. Empirical results for the distributions of b2 and√ b1. Biometrika, 60(3), 613–622.
5. *De Saussure F.* (1916), Course in General Linguistics. — New York, NY : Columbia University Press, 2011.
6. *Dinu, G., Lazaridou, A., Baroni, M.* (2014), Improving zero-shot learning by mitigating the hubness problem. arXiv preprint arXiv:1412.6568.
7. *Firth J. R., Strevens P. D.* (1930), The tongues of men and speech. — 1968.
8. *Firth J. R.* (1957), A synopsis of linguistic theory 1930–1955. Studies in Linguistic Analysis (Oxford: Philological Society): 1–32. Reprinted in F. R. Palmer, ed. (1968). Selected Papers of J. R. Firth 1952–1959. London: Longman.
9. *Gutiérrez, E. D., Levy, R., & Bergen, B. K.* (2016), Finding non-arbitrary form-meaning systematicity using string-metric learning for kernel regression. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 2379–2388.
10. *Kutuzov A., Andreev I.* (2015), Texts in, meaning out: Neural language models in semantic similarity tasks for Russian, Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue” (Moscow, May 27–30, 2015), issue 14 (21), Moscow, RGGU.
11. *Kutuzov A., Kuzmenko E.* (2017), WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. In: Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, vol 661. Springer, Cham
12. *Kuznetsova A, Efremova T.* (1986), Dictionary of Russian morphemes: circa 52,000 words [Словарь морфем русского языка: около 52 000 слов] — Russkiy Yazyk, 1986.

13. *Levenshtein V. I.* (1966), Binary codes capable of correcting deletions, insertions, and reversals //Soviet physics doklady. — 1966. — T. 10. — № 8. — pp. 707–710.
14. *Leviant I., Reichart R.* (2015), Separated by an un-common language: Towards judgment language informed vector space modeling //arXiv preprint arXiv:1508.00106. — 2015.
15. *Lyashevskaya O., Grishina E., Itkin I., Tagabileva M.* (2009), Word-formation annotation of the Russian National Corpus — aims and methods [О задачах и методах словообразовательной разметки в корпусе текстов], Poljarnyj vestnik. — 2009. — T. 12. — pp. 5–25.
16. *Mantel N.* (1967), The detection of disease clustering and a generalized regression approach //Cancer research. — 1967. — T. 27. — № 2 Part 1. — pp. 209–220.
17. *Mikhalev A. B.* (2008), Psycholinguistic problems of phonaesthemes. Language being of humans and ethnic groups: cognitive and psycholinguistic aspects. [Психолингвистическая проблематика фонестемы. Языковое бытие человека и этноса: когнитивный и психолингвистический аспекты.], Proceedings of the 4th International Berezin Readings. — M.: INION RAN, MGLU, (14), 140–148.
18. *Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J.* (2013), Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems.
19. *Monaghan P., Shillcock R., Christiansen M., Kirby S.* (2014), How arbitrary is language? //Phil. Trans. R Soc. B. — 2014. — T. 369. — № 1651.
20. *Plungian V. A.* (2005), Why we make Russian National Corpus? [Зачем мы делаем Национальный корпус русского языка?], Otechestvennye Zapiski, 2.
21. *Řehůřek R., Sojka P.* (2010), Software framework for topic modeling with large corpora, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta.
22. *Segalovich I.* (2003), A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine, MLMTA, pp. 273–280.
23. *Shillcock R., Kirby S., McDonald S., Brew C.* (2001), Filled pauses and their status in the mental lexicon //ISCA Tutorial and Research Workshop (ITRW) on Disfluency in Spontaneous Speech. — 2001.
24. *Tikhonov A.* (2003), Russian derivational dictionary in 2 volumes: more than 145,000 words [Словообразовательный словарь русского языка: в двух томах: более 145 000 слов], Astrel, 2003.
25. *Turney P. D., Pantel P.* (2010), From frequency to meaning: Vector space models of semantics, Journal of artificial intelligence research. — 2010. — T. 37. — pp. 141–188.
26. *Zhuravlev A.* (1991), Sound and meaning [Звук и смысл], M.: Prosveschenie. — 1991. — T. 160.

WORD SENSE INDUCTION FOR RUSSIAN: DEEP STUDY AND COMPARISON WITH DICTIONARIES[1]

Lopukhin K. A. (kostia.lopuhin@gmail.com), Scrapinghub

Iomdin B. L. (iomdin@ruslang.ru), V. V. Vinogradov Russian Language Institute of the Russian Academy of Sciences, National Research University Higher School of Economics

Lopukhina A. A. (alopukhina@hse.ru), National Research University Higher School of Economics, V. V. Vinogradov Russian Language Institute of the Russian Academy of Sciences

The assumption that senses are mutually disjoint and have clear boundaries has been drawn into doubt by several linguists and psychologists. The problem of word sense granularity is widely discussed both in lexicographic and in NLP studies. We aim to study word senses in the wild—in raw corpora—by performing word sense induction (WSI). WSI is the task of automatically inducing the different senses of a given word in the form of an unsupervised learning task with senses represented as clusters of token instances. In this paper, we compared four WSI techniques: Adaptive Skip-gram (AdaGram), Latent Dirichlet Allocation (LDA), clustering of contexts and clustering of synonyms. We quantitatively and qualitatively evaluated them and performed a deep study of the AdaGram method comparing AdaGram clusters for 126 words (nouns, adjectives, and verbs) and their senses in published dictionaries. We found out that AdaGram is quite good at distinguishing homonyms and metaphoric meanings. It ignores disappearing and obsolete senses, but induces new and domain-specific senses which are sometimes absent in dictionaries. However it works better for nouns than for verbs, ignoring the structural differences (e.g. causative meanings or different government patterns). The Adagram database is available online: http://adagram.ll-cl.org/.

Key words: semantics, polysemy, text corpora, word sense induction, semantic vectors

[1] This research was supported by RSF (project No.16-18-02054: Semantic, statistic and psycholinguistic analysis of lexical polysemy as a component of Russian linguistic worldview). The authors would also like to thank students of the Higher School of Economics and Yandex School of Data Analysis for their help in annotating dictionary senses.

АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ ЗНАЧЕНИЙ СЛОВ ДЛЯ РУССКОГО ЯЗЫКА: ДЕТАЛЬНОЕ ИССЛЕДОВАНИЕ И СРАВНЕНИЕ СО СЛОВАРЯМИ

Лопухин К. А. (kostia.lopuhin@gmail.com), Scrapinghub

Иомдин Б. Л. (iomdin@ruslang.ru), Институт русского языка имени В. В. Виноградова РАН, Национальный исследовательский университет «Высшая школа экономики»

Лопухина А. А. (alopukhina@hse.ru), Национальный исследовательский университет «Высшая школа экономики», Институт русского языка имени В. В. Виноградова РАН

1. Introduction

Several linguists and psychologists have drawn into doubt the assumption that word senses are mutually disjoint and that there are clear boundaries between them. Many psycholinguistic studies have found evidence for processing differences between distinct meanings (homonyms) and related senses (polysemes) (Frazier and Rayner, 1990; Rodd et al., 2002; Beretta et al., 2005; Klepousniotou and Baum, 2007; MacGregor et al., 2015), which shows that related senses are not associated with processing penalties. Moreover, polysemy processing seems to depend on sense overlap—high-overlap metonymic senses are processed easier than moderate- and low-overlap, metaphoric senses (Klepousniotou 2002; Klepousniotou et al., 2008, 2012; MacGregor et al., 2015). Eye movement evidence of how people process polysemous words with metonymic senses suggests that instead of accessing a specific sense, language users initially activate a word's meaning that is semantically underspecified (Frisson 2009, 2015).

The problem of sense distinction has also been discussed by lexicographers. Some of them are skeptical about the view of word meanings as sets of discrete and mutually exclusive senses (Cruse 1995; Kilgarriff 1997; Hanks 2000). Kilgarriff (1997) claims that sense distinction is worthwhile only with respect to a task at hand, while Hanks (2000) calls into question the phenomenon of word senses, showing how different components of the meaning potential of the word are activated in different contexts. Furthermore, word senses descriptions in dictionaries depend on the consistency of lexicographers and their theoretical basis. Sense divisions may be influenced by personal preferences, as lexicographers traditionally distinguish 'lumpers' and 'splitters' among colleagues: those who tend to break up senses further and those who go for large, homonymic senses (Wilks, 1998: 276). One possible solution to the problem of sense distinction was proposed by Erk and colleagues (2013). They found that untrained annotators prefer to disambiguate words in a context in a non-binary manner. People are often inconsistent with disjoint sense partitions and are more comfortable with a graded scale. Thus, the authors proposed to describe word meanings in the form of graded judgments in the disambiguation task (see also (McCarthy et al., 2016) about the notion of 'partitionability').

In the field of word sense disambiguation (WSD), the question of sense granularity is one of the key issues as the performance of WSD algorithms crucially depends on the sense inventory used for disambiguation. Although dictionaries and thesauri are the first option that comes to mind, they differ in the words that they cover, and also in the word senses that they distinguish. It was shown that sense distinction in most dictionaries is often too fine-grained for most NLP applications (see (Navigli, 2009) for a survey). This problem especially holds for the WordNet thesaurus (Fellbaum, 1998) and WordNet-like lexical databases—these resources are criticized for their excessive granularity that is not really needed for NLP tasks (Navigli, 2006; Snow et al., 2007) and for the loss of domain-specific senses (Pantel and Lin, 2002). Thus automated word sense induction (WSI) techniques may help establish an adequate level of granularity for NLP application and serve as empirically grounded suggestions for lexicographers.

Word sense induction is a task of automatically identifying the senses of words in raw corpora, without the need for handcrafted resources (dictionaries, thesauri) or manually annotated data. Generally WSI takes the form of an unsupervised learning task with senses represented as clusters of token instances (Navigli, 2009; Navigli, 2012; Nasiruddin, 2013). WSI results are often used as an input to WSD systems (Van de Cruys and Apidianaki, 2011; Navigli and Vannella, 2013) which allows to achieve state-of-the-art results in unsupervised WSD (Panchenko et al., 2016). Another NPL issue that benefits from word sense induction is web search clustering (Kutuzov, 2014). Di Marco and Navigli (2013) proposed a novel approach to web search result clustering based on word sense induction which outperformed both web clustering and search engines. In the fields of linguistics and lexicography WSI was successfully applied to the task of novel sense detection, i.e. identifying words which have taken on new senses over time (Lau et al., 2012, 2014). WSI also provides data for the study of diachronic variation in word senses (Bamman and Crane, 2011).

In this paper, we present an extensive comparison of four word sense induction techniques of different types. We chose two Bayesian approaches—a Latent Dirichlet allocation topic model (LDA) and a vector based Adaptive Skip-gram (AdaGram) model, one feature-based approach that represents each instance as a context vector, then utilizes a clustering algorithm, and an approach that performs clustering of word2vec neighbours. We quantitatively and qualitatively evaluated these techniques and performed a deep study of the AdaGram method comparing AdaGram clusters for 126 words (nouns, adjectives, and verbs) and their senses in published dictionaries. We studied sense overlap and types of senses that can be distinguished distributionally and by means of lexicographic theories. The research was done for Russian, and this is the first extensive study of the WSI methods for the Russian language.

2. Methods

A substantial number of different approaches to WSI has been proposed so far. They can be subdivided into local algorithms that discover senses separately for each word and global algorithms that allow to determine senses by comparing them to the senses of other words (Navigli, 2009; Van de Cruys and Apidianaki, 2011; Nasiruddin, 2013). In this study we compare four algorithms, two local and two global: Latent Dirichlet allocation that uses topic modeling, context clustering, word2vec neighbours clustering, and AdaGram.

Latent Dirichlet allocation (LDA) posits that each context is a mixture of a small number of topics (senses) and that each word's occurrence is attributable to one of the context's senses. Traditionally, Latent Dirichlet allocation is used for topic modeling in documents: each word in the document is assumed to originate from some topic, and the document can be represented as a mixture of topics. In case of the word sense induction, LDA is applied to contexts of one word, where documents correspond to target word contexts, and topics correspond to target word senses. The number of topics for LDA must be fixed in advance, but there are non-parametric variations like hierarchical Dirichlet process (HDP) that allow variable numbers of topics per document. One drawback of LDA here is that the word contexts are much smaller (just 10–20 words) than the documents that LDA is usually applied to (at least 100–1000 words). Each sense is represented as words that have most weight in the topic. LDA was trained on contexts extracted from the ruWac corpus with 6 topics for each word. No sampling of contexts was performed, most words had at least 10 thousand contexts. The words in each context were additionally filtered: only those with a weight greater than 1.0 were left. This is the same weighting as was used in the context clustering method and is described below and in (Lopukhin and Lopukhina, 2016).

In the **word2vec neighbours** method, we took word vectors closest to the target word and clustered them using spherical k-means, and then merged close clusters. This method is based on two assumptions. The first one is that the word2vec vector of the polysemous word will capture the properties of all senses that are encountered in the corpus frequently enough. The second assumption is that each sense of the polysemous word has at least one monosemous word with a similar meaning that occurs in similar contexts and thus has a similar embedding. Both of these assumptions have their weaknesses. The first assumption does not hold for rare senses. If a word in one of its senses is used in a small number of contexts, the word vector will not capture its meaning. The second assumption causes even more trouble, as many senses will not have any reasonable synonyms that are used in similar contexts often enough. Still, this method is very efficient, easy to implement and produces reasonable results for many words. Senses are represented as words closest to the center of each cluster. The clustering method and sense merging are described in more detail in the description of the context clustering method below.

Context clustering represents contexts as dense vectors, taking a weighted average of word2vec vectors of individual words. Context vectors are clustered using spherical k-means, and then close clusters are merged. In more detail, in this study each context was represented as a weighted average of word2vec embeddings of 10 words before and after the target word. Weights were equal to the pointwise mutual information of contexts words (Lopukhin and Lopukhina, 2016) and allowed to give more weight to words that are more important for disambiguation. This method of context representation proved to be efficient for word sense disambiguation (Lopukhin and Lopukhina, 2016; Lopukhina et al., 2016). The spherical k-means method was used for clustering of the context representations. Spherical k-means is similar to regular k-means clustering, but uses cosine distance instead of euclidian distance, which is a preferable measure of closeness for representations based on word2vec embeddings (Mikolov et al., 2013). The k-means clustering requires fixing the number of clusters in advance. But the number of senses is clearly different for different words, and k-means clusters often converge to very close points. To overcome both of these problems, clusters whose centers were closer than a certain threshold were merged. Senses are represented as most informative context words for a given sense.

AdaGram is a non-parametric Bayesian extension of the Skip-gram method. It automatically learns the required number of representations for all words at desired semantic resolution (Bartunov et al., 2015). It is able to learn the vector embedding for each sense of the word, where the number of senses is adapted depending on the number and diversity of contexts for each word. AdaGram has an efficient online learning algorithm that learns sense vectors for all words simultaneously. In practice, training is p times slower than for word2vec Skip-gram algorithm, where p is the maximum number of senses for a word (hyperparameter set in advance, typically 10–20). The model was evaluated on word sense induction tasks of SemEval-2007 and 2010 (Bartunov et al., 2015: 8–9) and achieved results superior to other extensions of word2vec to multiple senses. Besides p, the most important hyperparameter of AdaGram is α that controls granularity of produced senses. Other hyperparameters, such as vector dimension and window size, have the same meaning as in word2vec Skip-gram method. AdaGram can perform word sense disambiguation using induced senses and represents senses with nearest neighbors. We extended the sense representation with context words that give most information about a particular sense and typical sense contexts, and developed a Python library that allows loading AdaGram models and performing disambiguation. AdaGram model was built for about 190,000 most frequent words. Mean number of senses across all words is just 1.4, but more frequent words have more senses: 5.1 for the first 1,000 words and 3.6 for the first 10,000. The model is available online: http://adagram.ll-cl.org/about.

All models were trained on a 2 billion token corpus combining the ruWac Internet corpus (Sharoff, 2006), a Russian online library lib.ru and the Russian Wikipedia. All words were lowercased and lemmatized, no stop-word removal was performed. The word2vec Skip-gram model for word2vec neighbours and context clustering was trained with vector dimension 1024, window 5 and minimal token frequency 100 (forming a vocabulary of about 190,000 words). The AdaGram model was trained with maximum number of senses $p = 10$, sense granularity $\alpha = 0.1$, vector dimension 300, window 5 and minimal frequency 100. AdaGram has lower vector dimensionality, but this is compensated by the fact that multiple vectors are learnt for most words.

3. Evaluation

WSI evaluation is particularly arduous because there is no easy way of comparing and ranking different representations of senses. In fact, all the proposed measures in the literature tend to favour specific cluster shapes and consistency of the senses produced as output. Here we apply two clustering measures—V-measure and adjusted Rand Index. Moreover, we qualitatively evaluated the obtained clusters and compared them with sense distinction in dictionaries.

3.1. Quantitative evaluation

For the quantitative evaluation of different WSI methods we compared induced senses with dictionary senses for 8 polysemous nouns and 10 polysemous verbs. For each word, 100–500 contexts were sampled from RuTenTen11 (Kilgarriff et al., 2004) and RNC corpora (http://ruscorpora.ru/en/) and labeled with dictionary senses from the Active Dictionary of Russian (Apresjan, 2014) by a human annotator. The methods assigned each context to one of the induced senses. Thus we obtained two different clusterings of contexts for each word: one by a human annotator and one by a WSI method, and used two different clustering similarity measures to compare them. We did not do a quantitative evaluation of the word2vec neighbours method as it lacks a natural disambiguation approach: senses are induced directly from word2vec embeddings without using contexts; only a qualitative evaluation (below) was performed.

Table 1. V-measure for the word sense induction task

	Nouns	Verbs	Average
LDA	0.16	0.10	0.13
Context clustering	**0.39**	**0.22**	**0.31**
AdaGram	0.33	0.18	0.26

Table 2. Adjusted Rand Index for the word sense induction task

	Nouns	Verbs	Average
LDA	0.12	0.02	0.07
Context clustering	**0.34**	**0.14**	**0.24**
AdaGram	0.25	**0.13**	0.18

V-measure is a harmonic average of homogeneity and completeness of clusters. It was used in the SemEval-2010 Word Sense Induction & Disambiguation competition (Manandhar et al., 2010), but was criticized for favoring clusterings with a large number of clusters. This is less of a problem for our evaluation as we cap the maximum number of clusters for all methods at 10. Still, it is important to use an evaluation metric that corresponds to human intuition of having a reasonable number of clearly distinct senses, so we additionally used adjusted Rand Index (ARI) (Hubert and Arabie, 1985). It does not have the abovementioned issue and was used by Bartunov and colleagues (2015) in the AdaGram evaluation.

The quantitative comparison shows that context clustering and AdaGram are clearly better than LDA for nouns and especially for verbs. Context clustering performs better than AdaGram in this test for both ARI and V-measure, especially for nouns, but this comparison is not entirely fair: hyperparameters of context representation for context clustering were specifically tuned during WSD evaluation in (Lopukhin and Lopukhina, 2016; Lopukhina et al., 2016) that were performed on a similar set of words, while AdaGram hyperparameters were left at their default values. Close senses were not merged for AdaGram, this could also improve V-measure and especially ARI.

3.2. Qualitative evaluation

We also performed a qualitative evaluation of these methods on 15 nouns: 7 polysemous, having 3–9 senses in the Active Dictionary of Russian, and 8 nouns that have just one sense in the dictionary, but at least 5 of them have new and slang meanings (e.g. *bomba* 'crib', 'sexually attractive woman' and *bajan* 'old joke'). All induced senses were divided into three groups by a human annotator. The first group represented quality senses: senses that have an intuitively clear meaning, even if they are more or less fine-grained than the dictionary senses, or are completely absent from the dictionary. The second group represented duplicate senses that did not have sufficient distinctions from other similar senses. The third group represented senses that were hard to interpret: either a mixture of several clearly distinct senses, or just uninterpretable sense descriptions. Therefore, an ideal WSI method would produce a large number of quality senses and minimal number of duplicate or hard to interpret senses. The average number of senses in each group for all studied methods is presented in Table 3.

Table 3. Average number of quality, duplicate and unclear senses for the four WSI method

	Quality senses	Duplicate senses	Hard to interpret
Word2vec neighbours	2.4	1.1	0.5
Context clustering	2.8	1.0	0.9
LDA	1.8	2.1	1.3
AdaGram	3.6	3.7	2.5

The two best methods according to this metric are AdaGram and context clustering. AdaGram produces the largest number of quality senses, while also having more duplicates and hard to interpret senses. Context clustering has fewer duplicates and hard to interpret senses, while still giving a high number of quality senses.

While AdaGram and context clustering use conceptually similar context representation (bag of word vectors), AdaGram has one computational advantage over the context clustering method: it learns sense vectors for all words simultaneously, while context clustering requires extracting contexts and clustering for each word separately. On one hand, this makes it much easier to change the algorithm and its hyperparameters, but on the other hand, AdaGram is able to produce sense vectors for all words in the corpus much faster. This is why we chose AdaGram for a deeper qualitative evaluation on more words.

3.3. AdaGram qualitative evaluation

First, we compared the average recall. We prepared a dataset of 51 nouns, 40 verbs and 35 adjectives with different ambiguity types—homonyms, words with metaphoric and metonymic senses, terms and frequent highly polysemous words (according to the Frequency Dictionary of Russian (Lyashevskaya and Sharoff, 2009)). For all these words we compared senses that are distinguished in four dictionaries (the Russian Language Dictionary (Evgenyeva, 1981–1984), the Explanatory Dictionary of Russian (Shvedova, 2007), the Large Explanatory Dictionary of Russian (Kuznetsov, 2014) and the Active Dictionary of Russian (Apresjan, 2014)) with clusters induced by AdaGram. A cluster was considered a hit if it represented only one dictionary sense: mixed or broader clusters were rejected. This part of the evaluation was performed by many annotators without overlap, so inter-annotator agreement is unknown. Overall, the average recall for nouns is higher than for adjectives and verbs and is lower in comparison with the Active Dictionary of Russian than with other dictionaries.

Table 4. Average number of senses discovered by AdaGram in comparison to dictionaries (recall)

	Apresjan, 2014	Kuznetsov, 2014	Evgenyeva, 1981–1984	Shvedova, 2007	Average
adjectives	0.44	0.72	0.68	0.66	0.62
nouns	0.50	0.70	0.72	0.74	0.69
verbs	0.35	0.61	0.68	0.71	0.61
Average	0.43	0.68	0.70	0.71	0.64

In order to compare the sets of senses induced by AdaGram and described by lexicographers, we performed a following experiment. 98 polysemic words (30 nouns, 38 adjectives, 30 verbs) were chosen from the Active Dictionary of Russian. Then we performed a manual evaluation of the AdaGram clusters (an example of the model's output is presented in the Appendix). The Active Dictionary of Russian was chosen because it uses a series of linguistic criteria to systematically distinguish between senses of a given word (called lexemes).

In many cases, AdaGram distinguishes less senses than the dictionary does. As it appears, AdaGram usually does not induce obsolete, obsolescent, vernacular, special etc. senses, e.g. *bort* 'a front lap' (of a jacket or coat: *bort pidžaka <sjurtuka>*), *balovat'* 'to horse around' (*Smotri ne baluj!*), *vstupit'* 'to come in' (*vstupit' na pomost* 'to mount a dais'), *žaba* as in *grudnaja žaba* 'cardiac angina'. For some words, most of the senses are quite rare and therefore ignored by AdaGram, e.g. all senses of the verb *axnut'* except for the first and direct one ('to gasp'): 'to go off' (*v nebe axnulo* 'boom went the sky), 'to hit smb' (*axnut' po skule*), 'to hit smth' (*axnut' kulakom po stolu* 'to thump a table'), 'to drop smth with a loud noise' (*axnut' printer ob pol* 'to flop the printer down'), 'to empty a glass' (*axnut' stakan vodki* 'to gulp down a glass of vodka'). This might be explained by the simple fact that these senses might not occur in the corpus at all, or occur very rarely.

More interestingly, AdaGram does not distinguish senses which differ in argument structure rather than in semantic components or domain, e.g. causative meanings: *gasit'* 'to extinguish' (*gasit' svet* 'to switch off the lights') and 'to be the cause of extinguishment' (*Dožd' gasit koster* 'The rain puts out the fire'); *brit'* 'to shave' (*On ne breet podborodok* 'He does not shave his chin') and 'to get shaved (by a barber)' (*On breet borodu v barberšope* 'He gets shaved in a barbershop'). Lexicalized grammatical forms of adjectives are not considered by AdaGram as specific senses, e.g. *bližajšij* ('the closest', a superlative form of *blizkij* 'close', but also 'near': *v blizhajšie dni* 'in the next few days') or *vysšij* ('the highest', a superlative form of *vysokij* 'high', but also 'higher': *vysšee obrazovanie*' 'higher education').

On the other hand, in some cases AdaGram offers more senses than the dictionary. First of all, these are proper names, e.g. *Blok* (a surname, literally 'a block, a pulley'), *Avangard* (a hockey team, literally 'advance guard'), *Vidnoe* (a town, literally 'smth visible'), *Groznyj* (the Russian tzar and the capital of Chechnya, literally 'menacing'), etc., which are normally excluded from explanatory dictionaries (at least in the Russian lexicographic tradition). More often, AdaGram distinguishes between groups of contexts referring to different domains. For example, it divides into two clusters the following sets of collocates of the word *babočka* 'a butterfly': (1) *motylek* 'a moth', *strekoza* 'a dragonfly', *porxat'* 'to flutter', *krylyško* 'a winglet', *roit'sja* 'to swarm', (2) *gusenica* 'a caterpillar', *kajnozojskij* 'Cainozoic', *nasekomoe* 'an insect', *češuekrylyj* 'lepidopterous', *dvukrulyj* 'dipterous'. Obviously, these are not two different senses of the word *babočka*, but rather two types of texts (fiction vs. non-fiction) where it apparently occurs in distinctly different contexts. Similarly, AdaGram postulates two meanings for the word *oružie* 'weapon' with contexts corresponding to wars vs. computer games, *brak* (civil vs. religious marriage), *anglijskij* 'English' (history books vs. sports). For the noun *graf,* apart from the mathematical sense ('a graph'), not listed in the Active Dictionary of Russian, AdaGram gives as many as four types of contexts corresponding to counts or earls in Russia, France, Britain and Western Europe in general, which the dictionary considers belonging to the same sense.

In many cases, AdaGram offers several clusters of contexts which do not overlap with the dictionary senses. For the adjective *vozdušnyj*, it offers two groups of contexts: (1) *vozdušnyj potok* 'air flow'*, vozdušnyj fil'tr* 'air filter'*, vozdušnyj nasos* 'air pump'*, vozdušnyj poršen'* 'air piston', (2) *vozdušnyj šarik* 'party balloon', *vozdušnyj poceluj* 'air kiss', *vozdušnaja figurka* 'a feathery figurine', *vozdušnoe plat'e* 'a vapory dress', which the dictionary subdivides into five different senses: 'consisting of air', 'happening in the air', 'using air', 'using the energy of the air', 'lightweight'. Party balloons, kisses and dresses are more likely to be referred to in fiction, while air filters, pumps and pistols are more charachteristic for technical prose, and clearly these genres have quite different classes of contexts.

Finally, there are some special senses found by AdaGram but not listed in the dictionary; apart from the aforementioned *graf* 'graph', consider *agent* 'chemical agent', *vint* 'hard disk' (apparently a shortening from *vinčester* < Winchester), *gorjačij* 'used for communication' (*gorjačaja linija, gorjačij telefon* 'hotline').

Our analysis shows that less distant senses are less likely to be distinguished by AdaGram, according to the following hierarchy: homonyms > senses belonging to different subgroups > senses beloging to the same subgroup > exploitations of the same sense (all these entities are systematically distinguished and marked in the Active Dictionary or Russian). It is also worth noting that metaphors are much more recognizable by AdaGram than metonymic shifts, which might also correspond to the way they are treated by native speakers. Although AdaGram distinguishes less senses than the Active Dictionary of Russian, the feedback we received from our annotators shows that they are often more satisfied with smaller sets of senses found by the former than with the fine-grained distinctions provided by the latter.

4. Conclusion

In this paper we have explored the question of word sense induction for Russian. We applied four methods with different underlying approaches—a Latent Dirichlet allocation topic model (LDA) a vector based AdaGram model, a feature-based context clustering method and an approach that performs clustering of word2vec neighbours. Quantitative evaluation performed for nouns and verbs showed that context clustering and AdaGram are better than LDA for nouns and much better for verbs. The overall qualitative evaluation of the interpretability of the obtained clusters revealed that the two best methods are AdaGram and context clustering. They produce the largest number of quality senses while word2vec neighbours and LDA performance is less powerful. This result can be explained by the fact that LDA has access to less information in this setup: it works only on contexts of each word individually, while the other methods have access to the whole corpus, either directly for AdaGram or indirectly via word2vec embeddings for other methods. Context clustering works better than word2vec neighbours because it uses the contexts too, while word2vec neighbours requires existence of monosemous neighbours.

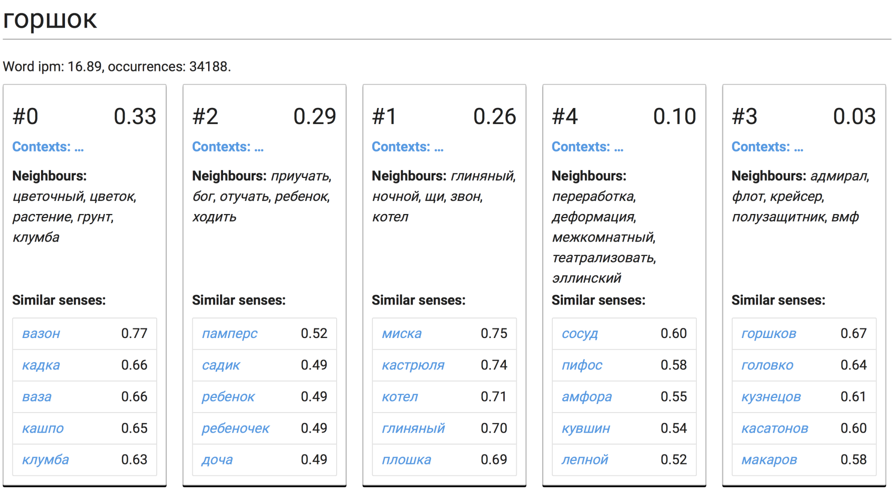
In a deeper study of AdaGram and its comparison with dictionaries, we found that the method performs consistently well for different parts of speech and induces overall 63% of senses that are distinguished by dictionaries. Moreover, AdaGram allows to get new and domain-specific senses that may not be included in domain-neutral lexical resources. The major limitation of the method is its inability to take syntactic information into account. The problem of sense discrimination by context is most evident for verbs. Although, AdaGram may not allow to solve the problem of the excessive granularity of lexical resources for NLP tasks (as it produces quite fine-grained clusters-senses), its clustering seems more corresponding to human intuition. And similarly to the conclusions from psycholinguistic experiments with ambiguous words, AdaGram distinguishes homonyms and metaphors better than more closely related senses.

The AdaGram database for Russian is available online (http://adagram.ll-cl.org/). The AdaGram method can be applied as a tool for lexicographers dealing with Russian—for sense induction from the large corpus and for novel sense detection. One of the AdaGram's possible application is the regular polysemy patterns detection which was discussed in (Lopukhina and Lopukhin, 2016). Besides, this instrument may help find patterns in big corpora that a human can not access.

One possible development of this study may be making context representations “aware of” the word order and the syntactic relations. This might allow distinguishing senses that are currently lumped together. This goal can be achieved either by corpus preprocessing (e.g. applying a syntactic parser), or by using richer context representations (e.g. moving from the bag of words to recurrent neural networks). Another possible development may be adjusting the methods to produce more distinct senses, and improving the sense presentations to make them clearer for the users.

Appendix

An example of the AdaGram model's output for the word *goršok* (# 0 'flower pot', #2 'potty' and 'clay pot', #1 'clay pot' and 'potty', #4 'clay pot', #3 'given name').



горшок

Word ipm: 16.89, occurrences: 34188.

#0 0.33

Contexts: ...

Neighbours: *цветочный, цветок, растение, грунт, клумба*

Similar senses:

вазон	0.77
кадка	0.66
ваза	0.66
кашпо	0.65
клумба	0.63

#2 0.29

Contexts: ...

Neighbours: *приучать, бог, отучать, ребенок, ходить*

Similar senses:

памперс	0.52
садик	0.49
ребенок	0.49
ребеночек	0.49
доча	0.49

#1 0.26

Contexts: ...

Neighbours: *глиняный, ночной, щи, звон, котел*

Similar senses:

миска	0.75
кастрюля	0.74
котел	0.71
глиняный	0.70
плошка	0.69

#4 0.10

Contexts: ...

Neighbours: *переработка, деформация, межкомнатный, театрализовать, эллинский*

Similar senses:

сосуд	0.60
пифос	0.58
амфора	0.55
кувшин	0.54
лепной	0.52

#3 0.03

Contexts: ...

Neighbours: *адмирал, флот, крейсер, полузащитник, вмф*

Similar senses:

горшков	0.67
головко	0.64
кузнецов	0.61
касатонов	0.60
макаров	0.58

References

1. *Apresjan Ju. D.* (ed.). (2014). Active Dictionary of Russian. Vol. 1 (A–B), Vol. 2 (V–G) [Aktivnyj slovar' russkogo jazyka. Tom 1 (A-B), tom 2 (V–G)]. Jazyki slavjanskih kul'tur, Moscow.
2. *Bamman, David and Gregory Crane.* (2011). Measuring historical word sense variation. Proceedings of the 2011 Joint International Conference on Digital Libraries (JCDL 2011), pp. 1–10, Ottawa, Canada.
3. *Bartunov, Sergey, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov.* (2015). Breaking sticks and ambiguities with adaptive skip-gram. Accessed February 15, 2017. https://arxiv.org/abs/1502.07257.
4. *Beretta, Alan, Robert Fiorentino, and David Poeppel.* (2005). The effects of homonymy and polysemy on lexical access: An MEG study. In: Cognitive Brain Research 24.1: 57–65.
5. *Cruse, D. A.* (1995). Polysemy and related phenomena from a cognitive linguistic viewpoint. In Philip Saint-Dizier and Evelyne Viegas, editors, Computational Lexical Semantics. Cambridge University Press, pages 33–49.

6. *Di Marco, Antonio and Roberto Navigli.* (2013). Clustering and diversifying Web search results with graph-based word sense induction. In: Computational Linguistics, 39(3):709–754.
7. *Erk, Katrin, Diana McCarthy, and Nick Gaylord.* (2013). Measuring word meaning in context. In: Computational Linguistics, 39(3):511–554.
8. *Evgenyeva A. P.* (ed.). (1981–1984), Russian Language Dictionary. Russian language, Moscow.
9. *Fellbaum, Christiane, editor.* (1998). WordNet, An Electronic Lexical Database. The MIT Press, Cambridge, MA.
10. *Frazier, Lyn and Keith Rayner.* (1990). Taking on semantic commitments: Processing multiple meanings vs. multiple senses. In: Journal of Memory and Language, 29:181–200.
11. *Frisson, Steven.* (2009). Semantic underspecification in language processing. In: Language and Linguistic Compass, 3, 111–127.
12. *Frisson, Steven.* (2015). About bound and scary books: The processing of book polysemies. In: Lingua, 157, 17–35.
13. *Hanks, Patrick.* (2000). Do word meanings exist? Computers and the Humanities. In: Senseval Special Issue, 34(1–2):205–215.
14. *Hubert, L. and Arabie, P.* (1985). Comparing partitions. Journal of classification, 2(1):193–218.
15. *Kilgarriff, Adam.* (1997). ‘I don't believe in word senses’. In: Computers and the Humanities, 31(2):91–113.
16. *Kilgarriff, Adam, Pavel Rychly, Pavel Smrz, and David Tugwell.* (2004). The Sketch Engine. In Euralex 2004. Proceedings, 105–116. Lorient, France.
17. *Klepousniotou, Ekaterini.* (2002). The processing of lexical ambiguity: Homonymy and polysemy in the mental lexicon. In: Brain and Language, 81:205–223.
18. *Klepousniotou, Ekaterini, & Baum, S. R.* (2007). Disambiguating the ambiguity advantage effect in word recognition: An advantage for polysemous but not homonymous words. In: Journal of Neurolinguistics, 20(1), 1–24.
19. *Klepousniotou, Ekaterini, Debra Titone, and Caroline Romero.* (2008). Making sense of word senses: The comprehension of polysemy depends on sense overlap. In: Journal of Experimental Psychology: Learning, Memory, and Cognition, 34(6):1,534–1,543.
20. *Klepousniotou, Ekaterini, Pike, G. B., Steinhauer, K., & Gracco, V.* (2012). Not all ambiguous words are created equal: An EEG investigation of homonymy and polysemy. In: Brain and Language, 123(1), 11–21.
21. *Kutuzov, Andrey.* (2014). Semantic clustering of Russian web search results: possibilities and problems. In: Russian Summer School in Information Retrieval. Springer International Publishing.
22. *Kuznetsov S. A.* (ed.). (1998), Large Explanatory Dictionary of Russian. Norint, St. Petersburg.
23. *Lau, Jey Han, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin.* (2012). Word sense induction for novel sense detection. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 591–601. Association for Computational Linguistics.

24. *Lau, Jey Han, Paul Cook, Diana McCarthy, Spandana Gella, and Timothy Baldwin.* (2014). Learning Word Sense Distributions, Detecting Unattested Senses and Identifying Novel Senses Using Topic Models. In Proceedings of ACL, 259–270. Baltimore, Maryland, USA.
25. *Lopukhin, Konstantin and Anastasiya Lopukhina.* (2016). Word Sense Disambiguation for Russian Verbs Using Semantic Vectors and Dictionary Entries. In: Computational Linguistics and Intellectual Technologies. No. 15. P. 393–405.
26. *Lopukhina, Anastasiya and Konstantin Lopukhin.* (2016). Regular polysemy: from sense vectors to sense patterns. In: The 26th International Conference on Computational Linguistics (COLING 2016). P. 19–23.
27. *Lopukhina, Anastasiya, Konstantin Lopukhin, Boris Iomdin, and Grigory Nosyrev.* (2016). The Taming of the Polysemy: Automated Word Sense Frequency Estimation for Lexicographic Purposes, in: Proceedings of the XVII EURALEX International Congress. Lexicography and Linguistic Diversity (6–10 September, 2016). Tbilisi: Ivane Javakhishvili Tbilisi State University.
28. *Lyashevskaya, Olga N., and Serge A. Sharoff.* (2009). Frequency dictionary of modern Russian based on the Russian National Corpus. Moscow.
29. *MacGregor, L. J., Bouwsema, J., & Klepousniotou, E.* (2015). Sustained meaning activation for polysemous but not homonymous words: Evidence from EEG. In: Neuropsychologia, 68, 126–138.
30. *Manandhar, S., Klapaftis, I. P., Dligach, D., and Pradhan, S. S.* (2010). SemEval-2010 task 14: Word sense induction & disambiguation. In: International workshop on semantic evaluation (SemEval), pp. 63–68.
31. *McCarthy, Diana, Marianna Apidianaki, and Katrin Erk.* (2016). Word sense clustering and clusterability. In: Computational Linguistics, Vol. 42, No. 2, Pages: 245–275.
32. *Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey and Dean.* (2013) Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26 (NIPS 2013), 3111–3119.
33. *Nasiruddin, Mohammad.* (2013). A state of the art of word sense induction: A way towards word sense disambiguation for under-resourced languages. arXiv preprint arXiv:1310.1425.
34. *Navigli, Roberto.* (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 105–112. Association for Computational Linguistics, USA.
35. *Navigli, Roberto.* (2009). Word sense disambiguation: A survey. ACM Computing Surveys, 41(2):1–69.
36. *Navigli, Roberto.* (2012). A quick tour of word sense disambiguation, induction and related approaches. In: International Conference on Current Trends in Theory and Practice of Computer Science. Springer Berlin Heidelberg.
37. *Navigli, Roberto, and Daniele Vannella.* (2013). Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application. In: Second Joint Conference on Lexical and Computational Semantics (* SEM). Vol. 2.

38. *Panchenko A., Simon J., Riedl M., Biemann C.* (2016). Noun Sense Induction and Disambiguation using Graph-Based Distributional Semantics. In: Proceedings of the 13th Conference on Natural Language Processing (KONVENS). Bochum, Germany.
39. *Pantel, Patrick, and Dekang Lin.* (2002). Discovering word senses from text. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM.
40. *Rodd, Jennifer, Gareth Gaskell, and William Marslen-Wilson.* (2002). Making sense of semantic ambiguity: Semantic competition in lexical access. In: Journal of Memory and Language 46.2: 245–266.
41. *Sharoff, S.* (2006). Creating general-purpose corpora using automated search engine queries. In Baroni and Bernardini, pp. 63–98.
42. *Shvedova N. Yu.* (2007), Explanatory Dictionary of Russian, Moscow.
43. *Snow, Rion, Sushant Prakash, Dan Jurafsky, and Andrew Y. Ng.* (2007). Learning to merge word senses. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 1005–1014. Prague, Czech Republic.
44. *Van de Cruys, Tim, and Marianna Apidianaki.* (2011). Latent semantic word sense induction and disambiguation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics.
45. *Wilks Y.* (1998). Senses and texts. In: Computational linguistics and Chinese language processing. V. 3. No. 2.

TESTING FEATURES AND METHODS IN RUSSIAN PARAPHRASING TASK

Loukachevitch N. V. (louk_nat@mail.ru),
Shevelev A. S. (alex.shevelev@hotmail.com),
Mozharova V. A. (joinmek@rambler.ru)

Lomonosov Moscow State University, Moscow, Russia

In this paper we study several groups of features and machine learning methods in the shared task on Russian paraphrasing organized in 2016. We use four groups of features: string-based features, information-retrieval features, part-of-speech features and thesaurus-based features and compare three machine learning methods: SVM with linear and RBF kernels, Random Forest and Gradient Boosting. In our experiments, the best results were obtained with the Random Forest classifier with parameter tuning and using all groups of features. The results of Gradient Boosting with parameter tuning were slightly worse.

Keywords: paraphrasing, semantic similarity, machine learning, thesaurus

ИССЛЕДОВАНИЕ ПРИЗНАКОВ И МЕТОДОВ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ ПАРАФРАЗ ДЛЯ РУССКОГО ЯЗЫКА

Лукашевич Н. В. (louk_nat@mail.ru),
Шевелев А. С. (alex.shevelev@hotmail.com),
Можарова В. А. (joinmek@rambler.ru)

МГУ им. М. В. Ломоносова, Москва, Россия

Ключевые слова: парафразы, семантическое сходство, машинное обучение, тезаурус

1. Introduction

Accounting paraphrases and synonyms is crucially important for various natural language applications such as machine translation (Marton et al., 2009), information retrieval and question answering (Fader et al., 2013), text summarization (Nenkova, McKeown, 2012; Loukachevitch, Alekseev, 2012), document clustering (Vossen et al., 2014), plagiarism measuring (Clough et al., 2002), etc.

Data for paraphrase detection can be found in synonym dictionaries, thesauri such as WordNet, or crowdsourced resources as Wikipedia. Also specialized databases with automatically collected paraphrases have been created (Dolan et al., 2004; Pavlick et al., 2015). Large text corpora can be processed to extract information on semantic similarity between words or expressions using similarity between their contexts (Przybyla et al., 2016). In practice paraphrase detection is based on large variety of sentence features (Kozareva, Montoyo 2006).

In this paper we describe results of exploiting several groups of features to detect paraphrased sentences in Russian. We are most interested in using semantic features calculated on the basis of RuThes thesarus (Loukachevitch, Dobrov, 2014). We also study several machine learning methods in this task: SVM, Random Forest, and Gradient Boosting. The evaluation is carried out on the data of the Russian Paraphraser corpus (Pronoza, Yagunova, 2016; Pivovarova et al., 2016).

2. Related Work

Most papers on English paraphrasing have been evaluated on Microsoft Research Paraphrase Corpus (Dolan et al., 2004), which comprises 5,081 paraphrase sentence pairs. The sentences pairs have been manually annotated into two classes: paraphrases or not. The data contain 67% positive examples of paraphrases and 33% of non-paraphrases. The data have been arbitrarily split into a training set containing 4,076 examples and a test set containing 1,725 examples. Evaluation of approaches to semantic textual similarity is also organized in the framework of SemEval conference (Agirre et al., 2016).

Most approaches to paraphrase detection exploit the following groups of features and combine them with machine learning methods (Kozareva, Montoyo 2006):

- various measures of word and character similarities, including length features, longest common sequence, n-gram overlap features, edit distances, machine translation similarities (BLUE, WER, TER, ROUGE-L etc.), information-retrieval measures (tf-idf, BM25), named entity similarity (Brychcın, Svoboda 2016);
- features of lexical differences between sentences including parts of speech tags, named entities, meaningful words (Pronoza, Yagunova, 2015a);
- syntactic features based on similarity between dependency trees;
- semantic measures based on WordNet conceptual structure (Mihalcea et al. 2006; Fernando, Stevenson, 2008);
- corpus-based similarities using classical distributional vectors or distributed representations of words learned by neural networks on a large text corpus (Przybyla et al., 2016);

Last successful approaches in paraphrase detection combine neural networks, comparison of dependency trees and semantic measures based on WordNet similarity (Rychalska et al., 2016; Brychcın, Svoboda 2016).

The previous work on semantic approaches for paraphrasing in Russian includes the work by Dobrov and Pavlov (2010) who studied the contribution of synonyms described in the Socio-political thesaurus for Russian news document clustering. With this aim, they created the conceptual index where each concept contains all known synonyms for news texts. For evaluation, the collection of news documents from ROMIP (Russian Information Retrieval Seminar)[1] was used. They found that the use of the conceptual index improves the best achieved result of news clustering (if compared with clustering based on words in the text body and the header) by 5.5%.

Pronoza and Yagunova study (2015a) various factors of paraphrase detection on the Russian paraphrase corpus including shallow measures based on word or characters overlap, dictionary-based measures and distributional semantic measures based on finding context similarity between words in a text corpus. They experimented on the Russian paraphrase corpus containing 6,281 sentence pairs (1,482 precise, 3,247 loose and 2,209 non-paraphrases). Altogether more than 80 features of sentences were calculated and combined with the Gradient Boosting classifier. The similarity between synonyms in a dictionary was based on calculating the probability to meet the words in the same set of synonyms.

In 2016 the shared task on evaluation of methods for detecting Russian paraphrases has been organized (Pivovarova et al., 2016).

3. Russian Paraphrase Evaluation: Tasks, Data, Results

The main task in the evaluation was three-way classification of sentence pairs: precise, loose and non-paraphrases on the specially created Paraphraser corpus (Pivovarova et al., 2016). The task of binary classification was also considered: sentence pairs should be classified to paraphrases or non-paraphrases.

The participating teams should take a pair of sentences as an input and return the similarity class as a response. Participants could submit "standard" runs that utilize as training data only the ParaPhraser corpus and (or) manual dictionary resources, and "non-standard" runs that may use any other data. "Standard" and "non-standard" run have been evaluated separately.

The datasets were formed on the basis of news story headlines. The training collection contains about 7,000 sentence pairs. Each candidate pair was manually annotated by three native speakers with the use of crowdsourcing. The test dataset (Gold standard set) contains 1,924 sentence pairs.

Table 1. Russian Paraphrase Evaluation Dataset Statistics

Paraphrases	Training set	Gold standard set
Exact	1,662 (23%)	374 (19.4%)
Loose	3,644 (41%)	778 (40.4%)
Non-paraphrases	1,921 (36%)	772 (40.2%)
Total	7,227	1,924

The quality of submitted results has been assessed with Accuracy and macro F-measure. At present, the evaluation results are published only in the electronic form[2].

[1] http://www.romip.ru/

[2] http://www.paraphraser.ru/contests/result/?contest_id=1

4. Features for Paraphrase detection

For finding paraphrases, we use four groups of features and study results for three machine-learning methods (SVM, Gradient Boosting and Random Forest) in dependence of different parameters.

Table 2. The best results achieved at Russian Paraphrase Evaluation

Task	Accuracy	F-measure
Three-class, standard	59.01	56.92
Three-class, non-standard	61.81	58.38
Two-class, standard	74.59	80.44
Two-class, non-standard	77.39	81.10

The features include the following groups: string-based features, information-retrieval features, part-of speech features, and thesaurus-based features.

String-based features include features for two and three symbol N-grams, and for word one, two and three N-grams. For each type of N-grams, the string feature group comprises the following three features:

$$feature_1 = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

$$feature_2 = \frac{|S_1 \cap S_2|}{|S_1|}$$

$$feature_3 = \frac{|S_1 \cap S_2|}{|S_2|}$$

where S_1 is a set of character of word N-grams of Sentence 1; S_2 is a set of character of word N-grams of Sentence 2, $|S|$ is the number of elements in the set S.

Information-retrieval (IR) features include BM25 and IDF features computed on the train collection (Manning et al., 2008). The BM25 feature compares similarity of two sentences, using formula (*). The IDF features (Formula **) are calculated for the word difference between two sentences: maxIDF is the maximal idf for words in the sentence difference, sum IDF is the sum of all idf of words from the sentence difference. Calculating IDF, we suppose that the loss of frequent words in the difference between sentences may be not very meaningful.

$$BM25(S_1, S_2) = \sum_{i=1}^{n} IDF(w_i) * \frac{TF(w_i,S_2)*(k+1)}{TF(w_i,S_2)+k*(1-b+b*\frac{|S_2|}{avg})} \qquad (*)$$

$$IDF(w_i) = \log\frac{N-N(w_i)+0.5}{N(w_i)+0.5} \qquad (**)$$

where $TF(w_i, S)$ is the frequency of word w_i in sentence S, N is the number of sentences in the training collection, $N(w_i)$ is the number of sentences containing word w_i, $|S|$ is the length of a sentence in words, avg is the average length of a sentence in the collection, k and b are parameters, their standard variants ($k = 1.2$, $b = 0.75$) are used (Manning et al., 2008).

Part-of-speech (POS) features are binary features that indicate what parts of speech are found in the difference between sentences. Five part-of-speech features show the presence of nouns, verbs, adjectives, adverbs and all other functional parts of speech in sentence difference.

Thesaurus (Thes) features are calculated on the basis the RuThes thesaurus (Loukachevitch, Dobrov, 2014). They will be described in the next section.

5. Semantic (Thesaurus) Features for Paraphrase Detection

It is useful to use semantic information about synonyms and semantically related language units to detect similarities between phrases. With this aim, we utilize RuThes thesaurus (Loukachevitch, Dobrov, 2014). The publicly available version of the RuThes thesaurus, RuThes-lite 2.0, comprises 31.5 thousand concepts, 115 thousand Russian words and expressions[3]. RuThes is a linguistic ontology, hierarchical net of concepts. It has many similarities with the Princeton Wordnet (Fellbaum, 1998) structure, therefore approaches for calculating semantic similarity proposed for wordnets can be applied to RuThes also.

We calculated several lexical similarity measures proposed for Princeton WordNet. These measures exploit paths between concepts where words under comparison were assigned. The measures include Leacock-Chodorow measure (*Lch*), Lin measure (*Lin*), and Jiang-Conrath measure (*Jcn*) (Budanitsky, Hirst 1998).

The *Lch* measure estimates the similarity of two nodes by finding the path length between them in the is-a hierarchy. It is computed as:

$$sim_{lch} = -\log \frac{N_p}{2D}$$

where *Np* is the distance between nodes and *D* is the maximum depth in the taxonomy. The distance is calculated in nodes, that is the distance between synonyms is equal 1, and the distance between a node and its hypernym is equal 2. We used two variants of calculation of this measure: 1) using only hyponym-hypernym relations (Lch_1) and 2) using hyponym-hypernym and part-whole relations (Lch_2). In RuThes, the transitivity of part-whole relations is supported (Loukachevitch, Dobrov 2015), therefore multi-step paths of part-whole relations and their combination with hyponym-hypernym relations are also meaningful. In RuThes-lite, the maximum depth of the ontology accounting both types of relations is equal 14. The logarithm base is equal to 2*D*.

Other two measures are calculated on the basis of word probabilities and so called information content (IC). For every word the probability to meet this word in a corpus is calculated:

$$P(w) = \frac{Freq_w}{N}$$

where N is the size of a corpus in words. The probability of a concept is the sum of probabilities of all text entries assigned to this concept.

[3] http://www.labinform.ru/ruthes/index.htm

The information content of a concept is an estimate of how informative the concept is. It is supposed that frequently occurring concepts have low information content and rarely occurring concepts have high information content.

$$IC(c) = -\log(P(c))$$

In calculating information content, probabilities of all lower concepts in the hierarchy should be summed up. The *Lin* measure is calculated as follows:

$$sim_{lin} = \frac{2 \cdot IC(LCS(C_1, C_2))}{IC(C_1) + IC(C_2)}$$

where *LCS* is the least common subsume of *C*1 and *C*2.

$$sim_{jcn} = \frac{1}{IC(C_1) + IC(C_2) + 2 \cdot IC(LCS(C_1, C_2))}$$

For *Lin* and *Jcn* measures, two variants were also calculated: with and without accounting part-whole relations.

To estimate word frequencies for *IC* calculation, an additional news corpus was used. Therefore according to the evaluation rules, when we use the *Lch* measure, the run could be considered as standard. But when we use the *Lin* or *Jcn* similarity measures, these runs should be categorized as non-standard due to the use of the additional corpus.

Comparing sentences on the basis of thesaurus similarity, we use the approach proposed in (Fernando, Stevenson, 2008) that allows summing the similarity of a word in one sentence with several words from another sentence. Sentences in this approach are represented as binary vectors $\vec{a}$ and $\vec{b}$. The similarity between the sentences is calculated as follows:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a}W\vec{b}}{|\vec{a}||\vec{b}|}$$

where *W* is a square matrix of the calculated similarities between words and expressions found in both sentences.

Each w_{ij} in *W* represents the similarity of words w_i and w_j according to some lexical similarity measure. In our case the measures are symmetric, i.e. $w_{ij} = w_{ji}$ and the matrix is also symmetric. Diagonal elements represent self similarity and have the greatest values equal to 1.

Table 3. Matrix of *Lch* similarity for the example sentences

	Деми	*Мур*	*Украсть*	*Похитить*	*Одежда*
Деми	1	0	0	0	0
Мур	0	1	0	0	0
Украсть	0	0	1	0.7941	0
Похитить	0	0	0.7941	1	0
Одежда	0	0	0	0	1

As preprocessing, before thesaurus features calculating, sentences are lemmatized, function words are removed, numbers mentioned in sentences are substituted with corresponding words. Words not found in the thesaurus but met in both sentences have maximal similarity 1.

For example, if two sentences are considered:

(s1) *У Деми Мур украли одежду. (Demi Moor's clothes were stolen)*

(s2) *У Деми Мур похитили одежду. (Demi Moor's clothes were robbed)*

The matrix according the *Lch* measure is presented in Table 3. Words "*Деми*" and "*Мур*" are absent in the thesaurus but mentioned in both sentences. The different words *украсть* and *похитить* are linked with the hyponym-hypernym relation and have high semantic similarity according to the *Lch* measure.

6. Experiments and Results

Before comparison, all sentences were lemmatized and the part-of speech information was extracted for each word. In preliminary experiments, we chose Random Forest as a basic machine learning method. We used the implementation from scikit-learn package[4].

Table 4. Best results achieved using Random Forest learning (grid parameter tuning)

Features	Heldout set		Gold standard set	
	Accuracy	**F-measure**	**Accuracy**	**F-measure**
1) String-based	63.34	61.42	60.03	57.99
2) 1)+BM25	64.59	62.76	60.55	58.67
3) 2)+ Max idf	64.59	62.67	60.96	58.99
4) 3)+POS features	**65.76**	**63.87**	61.07	59.03
5) 4)+Thes$_{lch}$	65.35	63.56	61.48	59.33
6) 5)+Thes$_{jcn}$	65.28	63.35	**62**	**60.03**

The parameters of the method were tuned with `GridSearchCV`[5] function. This function generates candidates from a grid of parameter values specified with the `param _ grid` parameter. All the possible combinations of parameter values are evaluated and the best combination is retained. In our case for tuning parameters, the training set was subdivided into the cross-validation dataset and the heldout set. The parameters were tuned on the cross-validation dataset with the cross-validation technique and tested on the heldout set.

Table 5 contains the achieved results on the heldout set and the gold standard set for Random Forest with parameter tuning. It can be seen that string-based features allows obtaining the significant level of the result. If to compare with the Paraphrase evaluation results (Table 2), it can be noted that the string-based features with tuned

[4] http://scikit-learn.org/stable/index.html

[5] http://scikit-learn.org/stable/modules/grid_search.html

Random Forest overcome the results reported in the evaluation (Standard variant). Other groups of the proposed features gave further improvement of the results.

Table 5. Results achieved with the default parameters

Methods and Parameters	Features	Accuracy	F-measure
SVM linear Default parameters (C=1, penalty=L2)	String-based	59.82	56.54
	String-based+IR	60.86	57.49
	String-based+IR+POS	60.60	57.36
	String-based+IR+POS+Thes	**61.43**	58.10
SVM rbf Default parameters C=1	String-based	58.99	56.95
	String-based+IR	59.77	57.77
	String-based+IR+POS	59.82	56.72
	String-based+IR+POS+Thes	60.49	57.62

Random Forest
Default parameters
N-estimators=10
Min_samples_leaf=10
	String-based	54.88	52.61

	String-based+IR	57.38	54.76
	String-based+IR+POS	57.43	55.66
	String-based+IR+POS+Thes	56.65	54.60

Gradient Boosting
Default parameters
N_estimators = 100
min_samples_leaf = 1
max_depth = 3
learning_rate = 0.1
	String-based	59.56	57.55

	String-based+IR	59.51	57.95
	String-based+IR+POS	60.91	58.89
	String-based+IR+POS+Thes	60.86	**59.11**

Table 6. Results achieved with grid parameter tuning: SVM (linear, RBF), Gradient Boosting

Methods and Parameters	Features	Accuracy	F-measure
SVM linear Grid tuning, C=0.4,0.7, 0.2, 0.2 Penalty L2	1) String-based	59.92	56.71
	2) String-based+IR	60.86	57.52
	3) String-based+IR+POS	60.75	57.54
	4) String-based+IR+POS+Thes	61.64	58.52

SVM rbf
Grid tuning
C=1.5, 100, 70, 0.6
Gamma=0.01, 0.1
	1) String-based	59.25	57.29

	2) String-based+IR	57.38	54.72
	3) String-based+IR+POS	58.00	54.85
	4) String-based+IR+POS+Thes	59.61	57.32
Gradient Boosting Grid tuning	1) String-based	60.13	58.17
	2) String-based+IR	60.55	58.65
	3) String-based+IR+POS	61.56	59.05
	4) String-based+IR+POS+Thes	**61.93**	**59.92**

We experimented with different sets of the thesaurus features. The best result (BestOfThesaurus) in combination with features of other groups was obtained using four thesaurus features: two variants of similarity based on the *Lch* measure (with and without accounting part-whole relations) and two variants of similarity based on the *Jcn* measure (Run 6 in Table 4).

For each run, the parameters of Random Forest were tuned separately. The number_of_ esimators parameter were changed from 100 till 500, and the min-samples_ leaf parameter was equal to 15 or 20.

After obtaining the results with tuned Random Forest, we compared the results of other machine learning methods working with the same feature set. We considered SVM (linear kernel and *radial basis function* kernel (RBF)) and Gradient Boosting. All methods were compared in two main regimes: with default parameters (Table 5) and with grid-tuned parameters (Table 6).

From Table 5, we can see that linear SVM achieves the performance close to the best result in Accuracy, and Gradient Boosting Method is enough close to the best result in F-measure without any tuning.

Table 6 shows the performance of the SVM and Gradient Boosting methods on the same features with tuned parameters. For Linear SVM and Gradient Boosting, the results slightly improved (if compared with default values of parameters) but were not better than for Random Forest. The parameter tuning for the rbf variant of SVM did not allow achieving better results on the Gold Standard set than with default parameters.

It also can be seen that the value of C-parameter for Linear SVM was always less than the default value (1). The C parameter for rbf SVM behaves unstable changing from 0.6 till 100.

Conclusion

In this paper we studied several groups of features and machine learning methods in the shared paraphrasing task in Russian organized in 2016. We used four groups of features: string-based features, information-retrieval features, part-of-speech features and thesaurus-based features and compared three machine learning methods: SVM with linear and RBF kernels, Random Forest and Gradient Boosting.

In our experiments, the best results were obtained with the Random Forest classifier with parameter tuning and using all groups of features. Each group of features improved the performance of paraphrase detection. The results of Gradient Boosting with parameter tuning were slightly worse.

Acknowledgments

This work is supported byRussian Science Foundation grant № 16-18-02074.

References

1. *Agirre E., Banea C., Cer D., Diab M., Gonzalez-Agirre A., Mihalcea R., Wiebe J.* (2016), Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. Proceedings of SemEval, pp. 497–511.
2. *Afzal N., Wang Y., Liu H.* (2016), MayoNLP at SemEval-2016 Task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model, Proceedings of SemEval-2016, pp. 674–679.
3. *Brychcın T., Svoboda L.* (2016), UWB at SemEval-2016 Task 1: Semantic textual similarity using lexical, syntactic, and semantic information, Proceedings of SemEval-2016, pp. 588–594.
4. *Brockett, C., Dolan, W. B.* (2005), Support vector machines for paraphrase identification and corpus construction, Proceedings of the 3rd International Workshop on Paraphrasing, pp. 1–8.
5. *Budanitsky A., Hirst G.* (2006), Evaluating wordnet-based measures of lexical semantic relatedness, Computational Linguistics, Vol 32, №. 1, pp. 13–47.
6. *Clough P., Gaizauskas R., Piao S., Wilks Y.* (2002), METER: MEasuring TExt Reuse, Proceedings of the 40th Anniversary Meeting for the Association for Computational Linguistics (ACL-02), pp. 152–159.
7. *Dobrov B., Pavlov A.* (2010), Basic line for news clusterization methods evaluation, Proceedings of the 5-th Russian Conference RCDL-2010, pp.287–295.
8. *Dolan W. B., Quirk C., Brockett C.* (2004), Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources, Proceedings of the 20th International Conference on Computational Linguistics Coling-2004, Geneva, Switzerland.
9. *Fellbaum Ch* (ed.). (1998), WordNet: An Electronic Lexical Database, The MIT Press.
10. *Fader A., Zettlemoyer L. S., Etzioni O.* (2013), Paraphrase-Driven Learning for Open Question Answering, Proceedings of ACL- 2013, pp. 1608–1618.
11. *Fernando S., Stevenson M.* (2008), A semantic similarity approach to paraphrase detection, Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, pp. 45–52.
12. *Kozareva Z., Montoyo A.* (2006), Paraphrase identification on the basis of supervised machine learning techniques, Advances in natural language processing. Springer Berlin Heidelberg, 2006. pp. 524–533.
13. *Loukachevitch N., Dobrov B.* (2014), RuThes Linguistic Ontology vs. Russian Wordnets, Proceedings of Global WordNet Conference GWC-2014, pp. 154–162.
14. *Loukachevitch N., Dobrov B.* (2015), The Sociopolitical Thesaurus as a resource for automatic document processing in Russian, Terminology. Special issue Terminology across languages and domains, Vol. 21, N 2, pp. 238–263.
15. *Loukachevitch N., Alekseev A.* (2012), Summarizing News Clusters on the Basis of Thematic Chains, Proceedings of LREC-2012, pp.1600–1607.
16. *Manning, C. D., Raghavan, P., Schütze, H.* (2008), Introduction to information retrieval. Cambridge: Cambridge University Press.
17. *Marton Y., Callison-Burch C., Resnik P.* (2009), Improved statistical machine translation using monolingually-derived paraphrases, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing EMNLP-2009, pp. 381–390.

18. *Mihalcea R., Corley C., Strapparava C.* (2006), Corpus-based and Knowledge-based Measures of Text Semantic Similarity, Proceedings of the American Association for Artificial Intelligence (AAAI 2006).
19. *Nenkova A., McKeown K.* (2012), A Survey of Text Summarization Techniques. Mining Text Data Book, Springer US, pp. 43–76.
20. *Pavlick E., Rastogi P., Ganitkevitch J., Durme B., Callison-Burch Ch.* (2015), PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification, Proceedings of ACL-2015 and the 7th International Joint Conference on Natural Language Processing, Vol. 2, pp. 425–430.
21. *Pronoza, E., Yagunova, E.* (2015a), Low-Level Features for Paraphrase Identification, Mexican International Conference on Artificial Intelligence, Springer International Publishing. pp. 59–71.
22. *Pronoza E., Yagunova E.* (2016), Construction of a Russian paraphrase corpus: unsupervised paraphrase extraction, Information Retrieval, Springer International Publishing, pp. 146–157.
23. *Pivovarova L., Pronoza E., Yagunova E.* (2016), Shared Task on Sentence Paraphrase Detection for the Russian Language, http://www.paraphraser.ru/download/get?file_id=2
24. *Przybyla, P., Nguyen, N., Shardlow, M., Georgios K. Ananiadou, S.* (2016), NaCTeM at SemEval-2016 Task 1: Inferring sentence-level semantic similarity from an ensemble of complementary lexical and sentence-level features, Proceedings of the 10th International Workshop on Semantic Evaluation SemEval-2016, pp. 614–620.
25. *Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., Andruszkiewicz, P.* (2016), Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, CA, USA.
26. *Vossen, P., Rigau, G., Serafini, L., Stouten, P., Irving, F., van Hage, W. R.* (2014), NewsReader: recording history from daily news streams, Proceedings of LREC-2014, pp. 2000–2007.

DOMAIN-INDEPENDENT CLASSIFICATION OF AUTOMATIC SPEECH RECOGNITION TEXTS

Mescheryakova E. I. (e-meshch@yandex.ru),
Nesterenko L. V. (lyu.klimenchenko@gmail.com)

National Research University Higher School of Economics; DC-Systems, Moscow, Russia

Call centers receive large amounts of incoming calls. The calls are being regularly processed by the analytical system, which helps people automatically inspect all the data. Such system demands a classification module that can determine the topic of conversation for each call. Due to high costs of manual annotation, the input for this module is the automatically transcribed calls. Hence, the texts (=automatic transcription) used for classification contain ill-transcribed words which can probably influence the classification process. Another important point is that this module also has special requirements: it should be domain-independent and easy to setup. Document classification task always requires an annotated data set for classifier training, but it seems to be too costly to make an annotated training set for each domain manually. In this paper, we propose an approach to automatic speech recognition texts classification that allows the user avoiding full manual annotation and at the same time to control its quality.

Key words: document classification, document clustering, automatic speech recognition, noisy texts processing

ТЕМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ АВТОМАТИЧЕСКИ ТРАНСКРИБИРОВАННЫХ ТЕКСТОВ ЛЮБОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Мещерякова Е. И. (e-meshch@yandex.ru),
Нестеренко Л. В. (lyu.klimenchenko@gmail.com)

Национальный исследовательский университет Высшая школа Экономики; DC-Systems, Москва, Россия

1. Introduction

Customer contact centers or call centers have to deal with a large amount of calls every day, and it appears to be difficult to manage this data and do the analytics manually. The supervisors and managers of call centers are interested in getting detailed analytic reports on a daily basis, which means it should not demand much human involvement and should be done quickly, in other words, it should be done automatically.

Getting the information about popular topics of incoming calls requires an analytic system to have an option of text classification/clustering (in our case the automatically transcribed calls recordings). Some solutions for this task had been proposed in [Agarwal et. al. 2007], [Popova et al. 2014], but the problem of domain-independent classification remains open. Here we propose an approach to domain-independent automatic speech recognition (ASR) texts classification. Our approach to handling noisy Russian data (texts with ASR mistakes) outdoes the one proposed in [Popova et al. 2014] and the use of clustering for semi-automatic training set annotation seems to be a solution to domain-independent classification.

The paper is organized as follows. In section 2 we make an overview of some papers devoted to clustering and classification of short ASR texts. Next, in section 3 we describe special characteristics of the data—the ASR texts. In sections 4 and 5 we describe the pipeline we implemented for domain-independent ASR texts classification and present the results of evaluation. Section 6 contains the conclusions of our work.

2. Related Work

While short noisy texts like social media content are a hot topic in NLP nowadays [Subramaniam 2009], ASR texts do not receive much attention. The ASR systems performance is stated to be high; however, when applied to call-center data, ASR quality decreases because of the system's sensitivity to loud environment and low-quality equipment. Obviously, it results in more errors for the languages with rich morphology like Russian.

In [Agarwal et. al. 2007], besides an overview of types of noise in textual data and related NLP tasks, one can find a number of experiments describing how ASR mistakes affect the supervised classification results (SVM vs. multinomial naive Bayes, English Reuters texts). The observations are optimistic: with word error rate up to 40%, a classifier accuracy does not decrease significantly. In [Popova et al. 2014] authors compare manual text transcripts and automatically recognized texts (word error rate about 20–35%) clustering and make the same conclusions as suggested in [Agarwal et. al. 2007] about the effect of error rate on the clustering results. It is also claimed that stop words (manually gathered domain-specific list) removal and Latent Semantic Indexing improve clustering results (best-averaged result stated is F1-score=0.47 on k-means with a stop-list when LSI is not used). Another work [Popova, Krivosheeva, Korenevsky 2014] proposes a more sophisticated approach to automatic stop words list generation: a word is included in the stop list if its removal from all the documents increases the dissimilarity between documents related to different clusters and also decreases the dissimilarity between documents within the same cluster. The best averaged F-measure achieved is 0.57.

ASR texts processing is usually done via supervised classification or clustering into a fixed number of clusters. The former approach demands a large manually annotated collection and the latter usually requires determining the number of clusters by a human or a robust procedure of finding one. A possible workaround is a two-stage clustering method [Wang, Wu, Shao 2014] where hierarchical clustering is performed in a sliding window and the clusters are iteratively merged using the information gain measure.

The approach we are proposing here is both simple and effective. Clustering allows us to avoid full manual annotation and at the same time to control the annotation quality.

3. Data

For the experiments, we used the dataset of 1,370 automatically transcribed calls of an airlines call center (all texts are in Russian). In the following, we refer to automatically transcribed calls as *texts* or *documents*. The dataset was manually annotated according to 5 topics, which are *luggage, booking change, ticket return, flight status,* and *flight information* (see Table 1 for the distribution of the topics).

Table 1. Topics distribution in the collection

Topic	Documents
luggage	653
booking	288
return	257
status	74
flight info	98
Total	1,370

These texts are typically short (min=18, max=1,439, median=170 words) and contain mistakes of the ASR system. The proportion of incorrectly transcribed word forms in ASR results is typically about 10–40% depending on audio quality. Below we refer to the words that were incorrectly transcribed by the ASR system as *noise*. Table 2 shows some examples of noisy sentences.

Table 2. ASR transcription errors. The erroneous words are in **bold**

	ASR transcription examples	Correct transcription
1	spasibo za **nogti konja** (thanks for **the horse nails**)	spasibo za zvonok vsego dobrogo do svidanija (thanks for the call and good bye)
2	davajte **kot** bronirovanija vam nazovu let me tell you the booking **cat**	davajte **kod** bronirovanija vam nazovu let me tell you the booking **code**
3	nazovite nomer **brone** pozhalujsta tell me your **bookings** code please	nazovite nomer **broni** pozhalujsta (tell me your **booking** code please)
4	**drova zazhiganija** firewood ignition	**spasibo za ozhidanie** (thanks for waiting)
5	broni **junosheskogo truda** skazhet booking **teenage labour** say	broni ? ? skazhet/skazhite booking ? ? say
6	mne by na popozzhe **rjabina** for me a bit later **ashberry**	mne by na popozzhe ? for me a bit later ?

As one can see, there are two major problems here. The first one is words deletion, and we do not deal with it in this paper. The second one is incorrect word transcription, and we see it useful to distinguish between two types of such mistakes. First, the mistakes (like examples 4–6 in Table 2), for which one can hardly name the correct word form or find any regularity in its appearance in texts. The other type of ASR mistakes (like examples 1–4 in Table 2) are the words that seem to be very similar to the correct transcription and they always stand for the same original words or, to put it differently, they are regular. In Section 4.2 we discuss how different ways of text vectorization allow us to cope with such noise.

Another salient characteristic of our data is its dynamics. A typical call-center that we are dealing with gets thousands of calls every day, and all of them have to be categorized. The distribution of topics can change in time depending on many external factors, with new topics appearing and some of the old ones vanishing. That means, we can not train a classifier once and be satisfied with the result. The fact that the data we deal with can change significantly obliges us to keep our classifier up-to-date and retrain it when needed.

When we apply the approach described below to the data we get for a new call-center project, at the starting point we do not know whether it is related to bank industry, telecommunications or any other domain. Quick setup for a new call-center project is also desirable and it should not involve time-consuming gathering or/and editing keywords lists.

To sum up, all the characteristics of data determine requirements to our classification module: resistance to ASR errors, timely model re-training, domain independence, and quick setup.

4. Implementation: From Clustering to Classification

4.1. Texts preprocessing and vectorization

For the purposes of quick setup, our pipeline demands minimal preprocessing of the ASR results. Firstly, the texts are lemmatized[1] and the stop words are deleted. We use a standard stop words list consisting of highly frequent Russian words (such as functional words and pronouns) and an additional wordlist containing words typical to call-centers (e.g. *talk, speak, please, hello* etc.; we found that for our purposes 60 words are enough). This list does not include domain-specific words and can be applied to various contact centers; this reduces customization costs. We also do not try to filter or correct ASR errors as most of them are being filtered automatically during the document vectorization procedure.

Normalized texts are then vectorized via one of the usual NLP techniques: tf-idf [Pedregosa et al. 2011] or doc2vec [Mikolov et al. 2013]. In order to compare different ways of vectorization, we performed classification using Random Forest Classifier (RFC, [Breiman 2011]), Logistic Regression [Hosmer et al. 2013] and SVM

[1] We used Mystem morphological analyzer [Segalovich 2003].

Classifier [Steinwart, Christmann 2008] on the same dataset vectorized by tf-idf, doc2vec distributed memory and doc2vec distributed bag-of-words models (Table 3). During the cross-validation procedure, training and test document sets were vectorized by tf-idf separately from each other on each iteration. When building a tf-idf collection matrix, the following the document frequency thresholds appeared to be the optimal: a word was not included in the tf-idf vocabulary if it was found in less than 20 documents or more than 50% of the collection. As for the optimal doc2vec model parameters, we finally set the vector size to 400 and the word frequency threshold to 3, i.e. a word that is ignored if it occurs less than 3 times in the collection.

Table 3. Different classifiers performance with tf-idf and doc2vec vectorization

Classifier, vectorization	**F1-score**
RFC (100 trees), tf-idf(max_df=0.5, min_df=20)	0.85
Logistic Regression (C=1), tf-idf(max_df=0.5, min_df=20)	0.86
SVM (C=1), tf-idf(max_df=0.5, min_df=20)	0.84
RFC (100 trees), doc2vec (size=400, min_count=3, distributed memory)	0.65
RFC (100 trees), doc2vec (size=400, min_count=3, bag of words)	0.62
Logistic Regression (C=1), doc2vec (size=400, min_count=3, distributed memory)	0.43
Logistic Regression (C=1), doc2vec (size=400, min_count=3, bag of words)	0.31
SVM (C=1), doc2vec (size=400, min_count=3, distributed memory)	0.30
SVM (C=1), doc2vec (size=400, min_count=3, bag of words)	0.31

The experiments had shown that tf-idf approach is preferred over the doc2vec models. We chose tf-idf model for the sake of its good performance, simplicity, and interpretability. Firstly, this helps to ignore most ASR mistakes during the classification as their document frequency is usually below the threshold (examples of the filtered words are given in Table 4); secondly, ASR mistakes defined above (see Section 3) as regular mistakes, if frequent enough to be in tf-idf vocabulary, are supposed to improve clustering to some extent.

Table 4. Stop words filtered by their document frequencies. These are obviously non-regular ASR errors

Stop words	**English translation**
file, veselit, razdelno, globus, izlechit, travmaticheskij, programmka, paluba, arest, lishaj	fillet, to amuse, separately, globe, to cure, traumatic, programme, deck, arrest, shingles

4.2. Clustering and clusters merging

Despite the fact that nowadays one has a large number of well-known clustering methods to choose from, the main challenge is still to determine the optimal number of clusters for a dataset. The problem is usually solved by optimization techniques such as elbow method, silhouette method, etc. However, we can not stick to one criterion as we try to make a domain-independent classification module. On the one hand, we want to avoid human involvement when possible, on the other hand, however, it is desirable to have an option that allows to edit clustering results if necessary. We solve this problem in the following way: the documents are clustered into deliberately larger, than it presumably is, number of clusters, so that their homogeneity is certainly high, and then the clusters are merged according to their lexical similarity. The merging procedure can be done or supervised by a human.

After the K-means clustering procedure (k-means++ initialization, 15 clusters), the average cluster homogeneity was 0.61, which we found acceptable. Adjusted rand index, on the other hand, was only 0.18, and completeness = 0.31.

Every cluster can be described by a list of most frequent lemmas bigrams (Table 5). Clusters merging procedure is therefore quite trivial and stands on these lists pairwise similarity. We refer to the named sets of clusters that this procedure results in as calls topics. The calls topics were named taking into account the manual annotation labels set. We make this assumption in order to perform the final evaluation in terms of the classification problem.

As shown in Table 5, the lists of the bigrams do not include much noise. Because of the high quality of these wordlists, it becomes possible for a person to adjust the clustering results and/or to name the calls topics if necessary.

Table 5. Most frequent lemmas bigrams of clusters

cluster id	**bigrams**	**calls topic**
# 0	*Russian* salon samolet, summa izmerenie, sem'desyat santimetr, santimetr summa, damskij sumochka, ruchnoj klast', damskij sumka, sumka noutbuk, sto pyatdesyat, dopolnitel'nyj plata, bagazh kilogramm *English translation* plane cabin, summ dimension, seventy centimeter, centimeter summ, lady's bag, hand luggage, women's bag, luggage kilograms	luggage

cluster id	bigrams	calls topic
# 1	*Russian* moskva ekaterinburg, nol’ nol’, vylet nol’, predstavitel’ aviakompanija, izmenit sorok, dar’ja predstavitel’, nol’ izmenit’, ekaterinburg vylet, tridcat’ utro, nol’ utro, moskva vylet *English translation* moscow ekaterinburg, zero zero, departure zero, airline representative, change forty, darja (name) representative, zero change, ekaterinburg departure, thirty morning, zero morning, moscow departure	flight status
# 2	*Russian* bagazhnyj otdelenije, salon samoljot, bagazhnyj otsek, bagazh salon, bagazh kilogramm, sto pyatdesyat, summa izmerenie, provoz bagazh, sdat’ bagazh, besplatno norma, pyatdesyat santimetr *English translation* luggage section, plane cabin, luggage place, luggage cabin, luggage kilogram, hundred fifty, sum dimension, carriage luggage, claim luggage, free norm, fifty centimeter	luggage
# 5	*Russian* data vylet, izmenit’ data, kupit’ bilet, tysjacha rubl’, familija passazhir, vylet vylet, bilet izmenit’, vylet napravlenie, annulirovat’ bilet, novyj bilet, denezhnyj sredstvo, nevozvratnyj tarif *English translation* departure date, change date, buy ticket, thousand ruble, surname passenger, departure departure, booking change, departure direction, cancel booking, new booking, money, economy class	booking change

4.3. Classification of the new documents

The procedure described above yields a large decently annotated collection of documents that can be used as a training set for the further classification. The classifier (best results were shown by Logistic Regression with C = 25)[2] is trained on the clustering results and predicts cluster ids for the new documents. Then these labels are mapped to the calls topic names according to the clusters naming done at the previous stage, and, finally, these results are compared to the manual annotation (Table 1). We evaluated the classifier's performance on the same dataset (Table 1) by dividing it into the training set, which was used for clustering, and the test set.

Table 6 shows the best classifier performance for each topic. The overall result—weighted average precision, recall and F-measure—is given in Table 7. The weighting was performed according to the proportion of 'true' instances of the particular topic class.

Table 6. Logistic Regression evaluation

Topic	Precision	Recall	F1-score	Number of documents
luggage	0.96	0.90	0.93	125
booking change	0.83	0.37	0.51	65
ticket return	0.48	0.90	0.63	52
flight status	0.58	0.69	0.63	16
flight information	0.73	0.50	0.59	16

Table 7. Weighted performance measures

Weighted Precision	Weighted Recall	Weighted F1
0.80	0.74	0.74

As shown in Table 6, the largest calls class ('luggage') was classified very well. We explain this by low lexical similarity of these documents with the others. On the other hand, 'flight status' and 'flight information' are rather often confused, and we see their closeness as the reason for high FP rate of the former and FN rate of the latter. The overall results seem satisfactory given that we did not edit the results of clustering. In comparison to the supervised classification results (Table 3 for the tf-idf vectorization), where F1 achieved 0.86, the results of the classifier trained on semi-automatically annotated dataset are slightly lower but still adequate.

5. Conclusion

In this paper, we observed the problem of domain-independent classification of automatic speech recognition texts and proposed a solution that allows to avoid fully manual annotation of the documents collection. Our results show that using clustering techniques as an automatic training set annotation tool does not worsen the classification results greatly. We regard the described pipeline as an acceptable solution for the case when one cannot afford manual annotation of a large training set.

[2] We used TPOT Python module [Olson et al. 2016] to chose the optimal classifier configuration.

References

1. *Agarwal, Sumeet, et al.* (2007), How much noise is too much: A study in automatic text classification. Data Mining ICDM 2007. Seventh IEEE International Conference on. IEEE.
2. *Breiman, Leo* (2001), Random Forests. Machine Learning. 45 (1), pp. 5–32.
3. *Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp.* (2011), Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM review 53.2, pp. 217–288.
4. *Hosmer Jr., David W., Stanley Lemeshow, and Rodney X. Sturdivant* (2013), Applied logistic regression. Vol. 398. John Wiley & Sons.
5. *Mikolov, Tomas, et al.* (2013), Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems.
6. *Olson R. S., Urbanowicz R. J., Andrews P. S., Lavender N. A., La Creis Kidd, and Jason H. Moore* (2016), Automating biomedical data science through tree-based pipeline optimization. Applications of Evolutionary Computation, pages 123–137.
7. *Pedregosa et al.* (2011), Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825–2830.
8. *Popova S. et al.* (2014), Automatic speech recognition texts clustering. International Conference on Text, Speech, and Dialogue. — Springer International Publishing, pp. 489–498.
9. *Popova S., Krivosheeva T., Korenevsky M.* (2014), Automatic Stop List Generation for Clustering Recognition Results of Call Center Recordings. International Conference on Speech and Computer. Springer International Publishing, pp. 137–144.
10. *Rosenberg, Andrew, and Julia Hirschberg* (2007), V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. EMNLP-CoNLL. Vol. 7.
11. *Steinwart, Ingo, and Andreas Christmann.* (2008), Support vector machines. Springer Science & Business Media.
12. *Subramaniam L. V. et al.* (2009), A survey of types of text noise and techniques to handle noisy text. Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data. — ACM, pp. 115–122.
13. *Wang Y., Wu L., Shao H.* (2014), Clusters merging method for short texts clustering. Open Journal of Social Sciences. Vol. 2. — № 09, p. 186.
14. *Segalovich, Ilya.* (2003), A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine.

IDENTIFYING DISEASE-RELATED EXPRESSIONS IN REVIEWS USING CONDITIONAL RANDOM FIELDS

Miftahutdinov Z. Sh. (zulfatmi@gmail.com)[1],
Tutubalina E. V. (elvtutubalina@kpfu.ru)[1],
Tropsha A. E. (alex_tropsha@unc.edu)[1,2]

[1]Kazan Federal University, Kazan, Russia
[2]University of North Carolina, Chapel Hill, USA

As the as the volume of user-generated content in social media expands so do the potential benefits of mining social media to learn about patient conditions, drug indications, and beneficial or adverse drug reactions. In this paper, we apply Conditional Random Fields (CRF) model for extracting expressions related to diseases from patient comments. Our method utilizes hand-crafted features including contextual features, dictionaries, cluster-based and distributed word representation generated from unlabeled user posts in social media. We compare our CRF-based approach with deep recurrent neural networks and a dictionary-based approach. We examine different word embeddings generated from unlabeled user posts in social media and scientific literature. We show that CRF outperformed other methods and achieved the F_1-measures of 69.1% and 79.4% on recognition of disease-related expressions in the exact and partial matching exercises, respectively. Qualitative evaluation of disease-related expressions recognized by our feature-rich CRF-based approach demonstrates the variability of reactions from patients with different health conditions.

Key words: disease named entity recognition, opinion expressions, conditional random fields, CRF, information extraction

ВЫЯВЛЕНИЕ СВЯЗАННЫХ С ЗАБОЛЕВАНИЯМИ ВЫРАЖЕНИЙ ИЗ ОТЗЫВОВ ПАЦИЕНТОВ НА ОСНОВЕ МОДЕЛИ УСЛОВНЫХ СЛУЧАЙНЫХ ПОЛЕЙ

Мифтахутдинов З. Ш. (zulfatmi@gmail.com)[1],
Тутубалина Е. В. (elvtutubalina@kpfu.ru)[1],
Тропша А. Э. (alex_tropsha@unc.edu)[1,2]

[1]Казанский федеральный университет, Казань, Россия
[2]Университет Северной Каролины, Чапел-Хилл, США

Ключевые слова: извлечение информации, условные случайные поля, анализ мнений, CRF, извлечение эффектов лекарств, извлечение симптомов заболеваний

1. Introduction

The explosive growth of social media has provided millions of people with the opportunity to share their thoughts or observations related to their health and health care. Repositories of user discussions such as patient portals can be often freely accessed by researchers interesting in social media listening to gather valuable new information about new uses of existing medications, adverse drug reactions, or unknown benefits associated with taking the medications.

A recent trend in text mining research is to move from detecting mentions of genes, gene variants, chemical/drug names, species and other biological concepts towards the broader task of extracting actionable insights from user feedback [1, 14, 17]. Research papers and electronic health records (EHRs) have been the subject of many experimental and clinical studies over the past decade [10]. The task of mining biomedical information from social media instead of articles and EHRs is more challenging due to the informal writing style of a text. Patients who are authors of comments lack formal medical skills to describe observed symptoms and drug reactions as medical concepts. Therefore, there is a growing interest in using machine learning approaches to enhance extraction of medical concepts from social media posts. Applications of these methods include pharmacovigilance and drug repurposing, that focus on extraction of adverse drug reactions (ADRs) and novel drug indications, respectively [7].

Conditional Random Fields (CRF) [16] have been successfully applied to numerous named entity recognition (NER) tasks including recognition of persons and organizations [8, 32], opinion aspects [3, 21], opinion expressions [11], and chemical and medical concepts [12, 18, 20, 23, 34]. In this paper, we apply CRF for the extraction of expressions associated with disease type from social media posts. Disease type consists of (i) entities that specify the reason for taking the drug (e.g., a specific disease name or symptoms of a disease), (ii) outcomes that can be attributed to some action of a drug (e.g., ADRs), and (iii) other findings like patient history. We employ an annotated corpus named CADEC that consists of 1250 medical forum posts taken from AskaPatient.com [14], where each post was manually annotated with mentions of drugs and disease-related entities such as symptoms, ADRs, and clinical findings. We compare CRF with bidirectional Long Short Memory Network (LSTM) and Gated Recurrent Units (GRU) [4, 9] and show that CRF is superior to the alternative approaches. The results of this study suggest that text mining of voluntary patient reports in social media using advanced methods such as CRF could be used as a reliable approach to identifying relationships between diseases (or medical conditions) and drug effects.

2. Related Work

Extr wexplore action of opinion targets (also called aspects) and opinion expressions has been pursued by many researchers using frequency-based methods, unsupervised and supervised methods. Most of the current unsupervised models are based on modifications of Latent Dirichlet Allocation (LDA) [26, 31]. The former are mainly based on Hidden Markov Models [36] and CRF [3, 5, 13]. Recently, bidirectional recurrent neural networks have been shown to outperform CRF on NER tasks [11, 21]. Irsoy and Cardie [11] applied deep Recurrent Neural Networks (RNNs) to extract direct or expressive subjective expressions. Three-layer RNN outperforms CRF, semi-CRF and shallow RNN. Liu et al. [21] applied RNNs for aspect extraction from datasets about laptops and restaurants. RNNs based on pre-trained word embeddings outperformed feature-rich CRF-based models. We mark [29, 30] about active learning and transfer learning as possible directions for future work.

The state-of-the-art models for disease-related information extraction from the literature in the BioCreative V task are also based on CRF [18, 19, 23, 34, 35]. Commonly used features include words, part-of-speech tags, word shape features, syntactic relations, and dictionaries. There was also a report [35] showing that RNNs in the BioCreative V task achieved lower results than CRF. Li et al. [20] used RNNs on the BioCreative II GM corpus to extract gene mentions from abstracts. Jagannatha and Yu [12] applied RNNs to extract entities of disease and medication types in EHRs. In addition to studying diseases, a lot of attention in recent years turned to the problem of mining ADRs from social media. One of the first studies on this subject [17] analyzed user posts regarding six drugs from a health-related social network. Benton et al. [1] analyzed message boards to detect drug events using dictionaries and co-occurrence statistics. Freifeld et al. [6] employed a dictionary-based approach to detect mentions of ADRs in tweets. Several studies used CRF to extract the ADRs from tweets [27, 33].

Most relevant to our work were studies by Metke-Jimenez and Karimi [24]. They applied dictionary-based methods and CRF to identify ADRs from the CADEC corpus. For CRF features, they used bag of words, letter n-grams, and word shapes (e.g., if the token composed of uppercase letters). The CRF outperformed other methods and achieved F1-measure of 60.2% in exact matching exercise. Our work differs from the aforementioned reports in several ways: (i) we focus on all disease-related entities, not only ADRs, since it could be more valuable for finding potentially novel causal relations among diseases and drugs; (ii) we experiment with not only feature-rich CRF-based approach but also with bidirectional LSTM and GRU; (iii) we xplore different hidden layer sizes of RNNs; (iv) we use word embeddings trained both on social media and the scientific literature; (v) we analyze the results to explore the variation in levels of effects across different patient conditions.

3. Approach

We formulate the disease-related entity extraction as a sequence labeling problem. CRF [16] is one of the state-of-the-art methods that takes a sequence of tokens as an input, calculates the probabilities of the predefined labels and selects the one with the maximum probability. We view an opinion as a sequence of tokens and label these tokens using the BIO (**B**eginning **I**nside **O**utside) tagging scheme. We identify BIO tags at the document level.

3.1. Features

We use the following set of features for CRF:

- Word (w): the lemmatized word itself;
- Part-of-speech tag (pos): the part-of-speech tag of each word;
- Suffix and Prefix (sp): the suffixes and prefixes of each word up to 6 characters in length;
- Context (context): three groups of features (x, pos, dict) of two words backward and two words forward from the current word;
- Word Type (wtype): two binary features that indicate whether a current word is a negation (*no*, *not* or *'t*) and whether all characters are capitalized;
- Dictionary Look-up (dict): if we a match can be found in the text, we mark the match using the BIO scheme. For each of three dictionaries, the token has 3 binary features: is beginning of matched part, is in "tail" of matched part, is out of matched part;
- Cluster-based representation (b): the vector of each word described below;
- Word embeddings (emb): the real-valued vector of each word described below.

We have made the implementation of CRF available at the github repository[1].

3.2. Dictionaries

We use the following dictionaries:

1. Dictionary of terms from the Unified Medical Language System (UMLS) with six disease-related types (333,905 entries);
2. Manually validated dictionary of terms (D_{terms}) from UMLS with semantic types "Finding" and "Mental Process" (6,608 entries);
3. ADR lexicon adopted from [27] (13,676 entries);
4. Manually created dictionary of multiword expressions (D_{MWE}) (943 entries);
5. Drug names with synonyms from the Drugbank database[2] (57,879 entries).

UMLS is a repository of biomedical vocabularies developed by the US National Library of Medicine. We have used the 2016AA edition of UMLS[1]. We extracted 562,919 medical terms with synonyms from UMLS with the following semantic types: "Disease Or Syndrome", "Neoplastic Process", "Sign Or Symptom", "Congenital Abnormality", "Mental or Behavioral Dysfunction", and "Anatomical Abnormality". We filtered out entries that were non-English terms, stop-words or body parts. The manually created dictionary contains MWEs starting with *feel*, *able* or *ability* such as "feel tired", "able to relax", "ability to move". In addition to medical terminology, UMLS contains Consumer Health Vocabulary, where terms have semantic types "Finding" and "Mental Process". However, we found many non-relevant to diseases terms with these types (e.g., *born in Cuba, parents got divorced*). In order to filter non-relevant terms, we calculated the frequency of each term in the Health Dataset (described below).

[1] https://github.com/dartrevan/ChemTextMining

[2] https://www.drugbank.ca/

Then we manually selected terms with high frequencies and combined them with synonyms in our dictionary (e.g., *drop in blood pressure*, *breakthrough bleeding*, *increased body weight*).

3.3. Word representations and Unlabeled Data

We used two types of word representations: (i) cluster-based and (ii) distributed (also called word embeddings). We collected a large number of 2,607,505 unlabeled user comments from six resources (this collection is further referred to as the Health Dataset) to induce the word representations. The resources included webmd.com[3], askapatient.com[4], patient.info[5], dailystrength.org[6], drugs.com[7]; we also employed health product reviews from freely available Amazon dataset[8]. Duplicate texts were removed. Each comment was processed with the tokenizer and lowercased.

We used the Brown hierarchical clustering algorithm [2]. This algorithm partitioned all words into a set of 150 clusters[9]. Word embedding models represent each word using a single real-valued vector. Such representation groups together words that are semantically and syntactically similar [25]. We used word2vec from Gensim library[10] to train embeddings on the Health Dataset. We applied Continuous Bag of Words model with the following parameters: vector size of 200, the length of local context of 10, negative sampling of 5, vocabulary cutoff of 10. Below, we refer to our pre-trained vectors as HealthVec (93,526 terms). We also experimented with another published word vector PubMedVec (2,351,706 terms) trained on biomedical literature indexed in PubMed [28].

4. Evaluation and Experiments

In this section, we conduct experiments to demonstrate the effectiveness of our CRF-based approach. We first describe the experimental settings and baselines. We compare CRF to the baseline methods and analyze the effect of different features.

[3] http://www.webmd.com

[4] http://www.askapatient.com

[5] https://www.drugs.com

[6] https://dailystrength.org

[7] http://patient.info

[8] http://jmcauley.ucsd.edu/data/amazon

[9] https://github.com/percyliang/brown-cluster

[10] https://radimrehurek.com/gensim/

4.1. Experimental Settings

The implementation of CRF was based on the sklearn-crfsuite library[11]. We used WordNetLemmatizer and maximum entropy tagger from the nltk library[12]. Passive aggressive algorithm was used for updating feature weights to train CRF.

Dataset. We use the CADEC corpus [14] that annotated with Drug and Disease entities at the sentence level (1250 posts, 7,632 sentences, 101,486 words). The corpus consists of four predefined disease-related types: ADR (6,318 entities), Disease (283 entities), Symptom (275 entities), Findings (435 entities). Since entities of each type are highly imbalanced in the corpus, we join them into one Disease type. We also reduced entities that fully contained in a larger entity of the same type. The total numbers of Drug and Disease entities were 1,799 and 6,752, respectively. To evaluate our method, we employed 5-fold cross-validation.

Baseline Methods. We evaluated our model by comparing with two baseline methods:

1. A knowledge-based approach that relies on the use of the described dictionaries and based on the exact lookup.
2. Bidirectional RNNs, in particular, LSTM and GRU [4, 9].

Our implementation of the knowledge-based approach is based on the Apache UIMA Ruta[13]. In order to implement RNNs, we used the Keras library[14]. The architecture of our networks and parameters are similar to [12]. We used a standard LSTM or GRU with the tanh activation function on top of the embedding layer. The embedding layer is based on pre-trained word embeddings. Bidirectional RNN has two independent forward and backward chains and the output layer that combines them. We use 100-dimensional hidden layer for each RNN chain. Finally, the combination of RNN chains' outputs is fed into a fully connected layer with softmax activation. This layer computes probabilities for each of the Drug and Disease labels and the Outside label. In order to prevent neural networks from overfitting, dropout of 0.5 is used to manage the inputs and the softmax layer. We use categorical cross entropy as the objective function. The batch size is 32. We use Adam [15] with a learning rate of 0.01 and a gradient clipping of 5.0 to optimize the cost of our network. We use a maximum of 70 epochs to train each network.

4.2. Experiments

At the pre-processing step, we performed spell correction[15]. We computed recall (R), precision (P) and F_1-measure (F) in two variants: (i) exact matching following CoNLL evaluation [32] and (ii) partial matching described in [22]. We use both Drug

[11] https://sklearn-crfsuite.readthedocs.io

[12] http://www.nltk.org/

[13] https://uima.apache.org/ruta.html

[14] https://keras.io/

[15] http://norvig.com/spell-correct.html

and Disease entities and do not present results of extraction of drugs since CRF and RNN extracts 92% of annotations correctly and the NER problem simply does not present itself. The results of different methods and ablation experiments are shown in Table 1 and Table 2, respectively.

Table 1: 5-fold cross-validation of the proposed methods

	Exact			**Partial**		
Method	**P**	**R**	**F**	**P**	**R**	**F**
Dictionary-based approach	.503	.502	.494	.836	.546	.625
1-layer GRU, HealthVec	.661	.516	.579	.786	.820	.780
2-layer LSTM, HealthVec	.712	.617	.661	.802	.863	.809
2-layer GRU, uniformly distributed rand. embeddings	.554	.489	.519	.740	.712	.694
2-layer GRU, PubmedVec	.669	.614	.640	.818	.800	.783
2-layer GRU, HealthVec	.719	.619	.665	.795	.871	.809
3-layer LSTM, HealthVec	.718	.629	.670	.801	.872	.812
3-layer GRU, HealthVec	.735	.629	.678	.793	.876	.811
CRF, all features + HealthVec	.702	.680	.691	.852	.790	.794

Table 2: 5-fold cross-validation of CRF with different feature groups

	Exact			**Partial**		
Method	**P**	**R**	**F**	**P**	**R**	**F**
features: w, sp, pos, context, wtype, b, dict, HealthVec	.702	.680	.691	.852	.790	.794
features: w, sp, pos, context, b, dict, HealthVec	.701	.681	.690	.853	.789	.794
features: w, sp, pos, context, b, dict, PubMedVec	.667	.682	.674	.829	.815	.799
features: w, sp, pos, context, dict, b	.667	.677	.672	.828	.812	.796
features: w, sp, pos, context, dict	.664	.672	.668	.828	.812	.797
features: w, sp, pos, context, dict (w/o D_{terms} and D_{MWE})	.665	.667	.666	.829	.804	.793
features: w, sp, pos, context	.651	.631	.641	.817	.778	.772
features: w, sp, pos	.615	.601	.608	.810	.771	.764

The results in Table 1 lead to several observations. First, 3-layer GRUs provide the best results as compared to other networks. Second, CRF achieved the best results in the exact matching exercise over GRU due to CRF's capability of predicting a valid sequence of the output labels. Third, F scores of CRF increased from 69.1% to 79.4% in the partial matching as compared to exact exercise since boundaries of opinion expressions are hard to define. Finally, we investigated the effectiveness of CRF’s features. The dictionaries along with vectors HealthVec based on in-domain texts led to the most gain in performance of CRF.

5. Analysis of disease-related entities associated with distinct conditions

Although medical terminology is limited, there are a large number of language expressions to describe conditions. To illustrate the variety of phrases which patients use to describe symptoms or drug reactions, we present a comparative analysis of extracted expressions for seven health conditions. For our analysis, we used 143,244 reviews from drugs.com, where each review corresponds to a drug and a condition for treatment. The number of conditions was 558. CRF extracted 684,567 entities from texts. Then we excluded unigrams and phrases that associated with more than one condition. To discuss subjective feelings of illness or drug reactions, we manually selected MWEs that contain words “feel”, “felt” or “feeling”. We present some examples in Table 3.

Table 3. Examples of MWEs associated with medical conditions.

Condition	Multiword Expressions
Fibromyalgia	electric feeling, felt some joint tightness in my neck, sunburn feel from my arms and legs, feeling extremely disoriented, feeling like I want to sneeze, felt like worms crawling, feeling flare ups of fibromyalgia symptoms, feeling of nails being driven through my feet away, feel groggy and drop off to sleep
Birth control	felt like I was constantly getting stabbed, feel like I was gonna pass out from the pain, felt like I was being ripped open inside, feel like having a mini surgery for birth control, felt like my head was going to explode, feel like a typical bloated boat walking around, feel like someone has just died, feeling like I was going insane, feels like my body cannot handle additional hormones in my body
Weight loss	feel “in the mood” to eat anything, no longer feel prisoner to the world of sweets, feeling like I was intoxicated, feel drained sun up to sun down, feel overweight, feel starved, feel the effects of reduced appetite and cravings, not feel out of control eating, feel like my appetite is suppressed, feeling of a decreased appetite
Anxiety	feeling like a deer in the headlights, feel like the room was spinning, feel like i am losing my grip, felt a tremendous sense of fear, feel incredibly awkward in social situations, feeling high or disoriented or mentally clouded, feel slightly less coordinated, feel like the constant dialogue in my head, feel like my heart is beating harder, heart feels like it’s going to fly out of my chest, feel despair
Panic disorder	feel like my emotions are so flat, feel like i am a weak person, feelings of guilt, fog-like feeling, feeling of lethargic/less energy, feel my heart starting to race, feel depressed/stress/panic/anxiety, feel hot from the inside of my body, feeling overall hopeless, feel gloomy, choking throat feeling, feeling like I was suffocating

Condition	Multiword Expressions
Bipolar disorder	feel “zombie-ish”, feel irritable or obsessed over things, feel very angry or depressed, feel genuine happiness, feel positive and optimistic, feel over whelmed with grief, feel manic or depressive, feels like something foreign in your body, feels like I have bugs crawling inside my brain, feels like everything kind of moves
Depression	feel a little emotionally numb, feels like an electrical impulse going through my head, feel sad or bothered by little things, feeling “revved” at night, feeling of pressure behind the eyes, feeling like I wanted to scream, feel like my head is sometimes floating above my body, feeling of “disconnection” from my emotions

Several observations can be made based on results in Table 3. First, we observe differences in expressions associated with mental health disorders, i.e. anxiety, panic disorder, bipolar disorder, depression. Panic disorder affects the emotional health of a patient (e.g., *emotions are so flat*, *hopeless*, *emotional numbness*), while people with anxiety experience emotional symptoms related to feelings of fear (e.g, *felt scared, fearful thoughts, distracted by irrational fear*s). The authors with bipolar disorder experience euphoric mood (e.g., *feel positive and optimistic, feel genuine happiness*), while depressed people feel withdrawn from socializing and hobbies (e.g., *loss of interest in everything*). These examples demonstrate that social media posts contain variable information for NER. Second, women that take birth control pills describe ADRs such as abnormal pain (e.g., *constantly getting stabbed, being ripped open inside, gonna pass out from the pain*) more emotional than patients with fibromyalgia, where muscle pain and muscle spasms are symptoms of the disease (e.g., *muscles feel extremely “tight”, worms crawling*). Third, patients very often don’t know what they are troubled by and use creative writing. For example, “prisoner to the world of sweets” is used to rephrase the term “sweet craving”, “a deer in the headlights” describes a feeling of being frozen, “feel like someone has just died” is used to describe depression, and “constant dialogue in my head” refers to a cognitive process such as rumination. Therefore, there is a need to create domain-specific dictionaries and map informal expressions to medical terms. Finally, there are shared problems for all disorders, e.g. most common ADRs like allergic reactions and rash (e.g., *sunburn*), drug abuse (e.g, *intoxicated*), or lack of control (e.g., *out of control eating, “disconnection” from my emotions, prisoner in your body*).

Our analysis shows that existing resources can integrate MWEs from social media posts to increase understanding of experiences of personalized expressive and explorative writings by patients and create valuable resources for supervised methods using these unique insights.

6. Conclusion

In this paper, we have explored the task of recognizing opinion expressions in social media associated with diseases and drugs. We complied and harmonized user expressions from multiple resources to create a collection we termed the Health Dataset. We used Conditional Random Fields (CRF) and implemented a variety of features based on contextual information, dictionaries, and word representations. We demonstrated the superiority of CRF as compared to a dictionary-based method and recurrent neural networks. We have also demonstrated the variability in emotional level of expressions depending on the type of patient conditions. Our analysis confirmed the need for qualitative methods to interpret informal disease-related expressions and map them onto medical terms. In addition to drug indications and adverse effects, we also plan to annotate beneficial effects, which could lead to the discovery of previously unknown drug effects and new drug repurposing hypotheses. Additional studies are needed to investigate if such effects may be a result of medication usage in combination with other factors such as life style or food. In future studies, we also plan to create and manually annotate a corpus of user reviews about medications, written in Russian. In summary, continuous advancement and improvement in the accuracy of text mining approaches applied to patient reports in social media will have plausible impact in several areas including pharmacovigilance (especially, for new drugs), drug repurposing, and understanding drug effects in the context of other factors such as concurrent use of other drugs, diet, and life style.

Acknowledgments

Work on problem definition, survey of related work, recurrent networks, dictionaries and qualitative analysis of results was carried out by Elena Tutubalina and supported by the Russian Science Foundation grant no. 15-11-10019. Other parts of this work were performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

1. *Benton A., Ungar L., Hill S., Hennessy S., Mao J., Chung A., Holmes J. H.* (2011), Identifying potential adverse effects using the web: A new approach to medical hypothesis generation, Journal of biomedical informatics, Vol. 44(6), pp. 989–996.
2. *Brown P. F., Desouza P. V., Mercer R. L., Pietra V. J. D., Lai J. C.* (1992), Class-based n-gram models of natural language, Computational linguistics, 18(4), pp. 467–479.
3. *Chernyshevich M.* (2014), IHS R&D Belarus: Cross-domain extraction of product features using conditional random fields, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 309–313.
4. *Cho K., Van Merriënboer B., Bahdanau D., Bengio Y.* (2014), On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259.
5. *Choi Y., Cardie C.* (2010), Hierarchical sequential learning for extracting opinions and their attributes, Proceedings of the ACL 2010 conference short papers, pp. 269–274.

6. *Freifeld C. C., Brownstein J. S., Menone C. M., Bao W., Filice R., Kass-Hout T., Dasgupta N.* (2014), Digital drug safety surveillance: monitoring pharmaceutical products in twitter, Drug safety, 37(5), pp. 343–350.
7. *Deftereos S. N., Andronis C., Friedla E. J., Persidis A., Persidis A.* (2011), Drug repurposing and adverse event prediction using high-throughput literature analysis, Wiley Interdisciplinary Reviews: Systems Biology and Medicine, 3(3), pp. 323–334.
8. *Gareev R., Tkachenko M., Solovyev V., Simanovsky A., Ivanov V.* (2013), Introducing baselines for Russian named entity recognition, Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, pp. 329–342.
9. *Hochreiter S., & Schmidhuber, J.* (1997), Long short-term memory. Neural computation, Vol. 9(8), pp. 1735–1780.
10. *Huang C. C., Lu Z.* (2016), Community challenges in biomedical text mining over 10 years: success, failure and the future, Briefings in bioinformatics, 17(1), pp. 132–144.
11. *Irsoy O., Cardie C.* (2014), Opinion Mining with Deep Recurrent Neural Networks. EMNLP, pp. 720–728.
12. *Jagannatha A. N., Yu H.* (2016). Bidirectional RNN for Medical Event Detection in Electronic Health Records, Proceedings of NAACL-HLT, pp. 473–482.
13. *Jakob N., Gurevych I.* (2010), Extracting opinion targets in a single-and cross-domain setting with conditional random fields. Proceedings of the 2010 conference on empirical methods in natural language processing, pp. 1035–1045.
14. *Karimi S., Metke-Jimenez A., Kemp M., Wang C.* (2015), Cadec: A corpus of adverse drug event annotations, Journal of biomedical informatics, Vol. 55, pp. 73–81.
15. *Kinga D., Adam J. B.* (2015), A method for stochastic optimization, International Conference on Learning Representations (ICLR).
16. *Lafferty J., McCallum A., Pereira F.* (2001), Conditional random fields: Probabilistic models for segmenting and labeling sequence data, Proceedings of the eighteenth international conference on machine learning, ICML, Vol. 1, pp. 282–289.
17. *Leaman R., Wojtulewicz L., Sullivan R., Skariah A., Yang J., Gonzalez G.* (2010), Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks, Proceedings of the 2010 workshop on biomedical natural language processing, pp. 117–125.
18. *Lee H. C., Hsu Y. Y., Kao H. Y.* (2015), An enhanced CRF-based system for disease name entity recognition and normalization on BioCreative V DNER Task, Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, pp. 226–233.
19. *Li D., Afzal N., Mojarad M. R., Elayavilli R. K., Liu S., Wang Y., Liu H.* (2015), Resolution of chemical disease relations with diverse features and rules, Proceedings of the fifth BioCreative challenge evaluation workshop, pp. 280–285.
20. *Li L., Jin L., Huang D.* (2015), Exploring recurrent neural networks to detect named entities from biomedical text, Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 279–290.
21. *Liu P., Joty S. R., Meng H. M.* (2015), Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings, Proceedings of EMNLP, pp. 1433–1443.

22. *Loukachevitch N., Blinov P., Kotelnikov E., Rubtsova Y., Ivanov V., Tutubalina E.* (2015), SentiRuEval: testing object-oriented sentiment analysis systems in Russian, Proceedings of International Conference Dialog, Vol. 2, pp. 3–13.
23. *Lu Y., Ji D., Yao X., Wei X., Liang X.* (2015), CHEMDNER system with mixed conditional random fields and multi-scale word clustering, Journal of cheminformatics, Vol. 7(1).
24. *Metke-Jimenez A., Karimi S.* (2015), Concept extraction to identify adverse drug reactions in medical forums: A comparison of algorithms, arXiv preprint arXiv:1504.06936.
25. *Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J.* (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, pp. 3111–3119.
26. *Moghaddam S., Ester M.* (2011), ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews, Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 665–674.
27. *Nikfarjam A., Sarker A., O'Connor K., Ginn R., Gonzalez G.* (2015), Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features, Journal of the American Medical Informatics Association, pp. 1–11.
28. *Pyysalo S., Ginter F., Moen H., Salakoski T., Ananiadou S.* (2013), Distributional semantics resources for biomedical text processing, Proceedings of Languages in Biology and Medicine.
29. *Qu L. et al.* (2016) Named Entity Recognition for Novel Types by Transfer Learning //arXiv preprint arXiv:1610.09914.
30. *Stanovsky G., Gruhl D., Mendes P. N.* (2017) Recognizing Mentions of Adverse Drug Reaction in Social Media Using Knowledge-Infused Recurrent Models.
31. *Titov I., McDonald R. T.* (2008), A Joint Model of Text and Aspect Ratings for Sentiment Summarization, ACL, Vol. 8, pp. 308–316.
32. *Tjong K., Sang E. F., De Meulder F.* (2003), Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, Vol. 4, pp. 142–147.
33. *Wang W.* (2016), Mining adverse drug reaction mentions in twitter with word embeddings. Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing.
34. *Wei C. H., Peng Y., Leaman R., Davis A. P., Mattingly C. J., Li J., Lu Z.* (2015), Overview of the BioCreative V chemical disease relation (CDR) task, Proceedings of the fifth BioCreative challenge evaluation workshop, pp. 154–166.
35. *Wei Q., Chen T., Xu R., He Y., Gui L.* (2016), Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks, Journal of Biological Databases and Curation.
36. *Wong T. L., Bing L., Lam W.* (2011), Normalizing web product attributes and discovering domain ontology with minimal effort, Proceedings of the fourth ACM international conference on Web search and data mining, pp. 805–814.

DETECTING INTENTIONAL LEXICAL AMBIGUITY IN ENGLISH PUNS

Mikhalkova E. V. (e.v.mikhalkova@utmn.ru),
Karyakin Yu. E. (y.e.karyakin@utmn.ru)

Tyumen State University, Tyumen, Russia

The article describes a model of automatic analysis of puns, where a word is intentionally used in two meanings at the same time (the target word). We employ Roget's Thesaurus to discover two groups of words, which, in a pun, form around two abstract bits of meaning (semes). They become a semantic vector, based on which an SVM classifier learns to recognize puns, reaching a score 0.73 for F-measure. We apply several rule-based methods to locate intentionally ambiguous (target) words, based on structural and semantic criteria. It appears that the structural criterion is more effective, although it possibly characterizes only the tested dataset. The results we get correlate with the results of other teams at SemEval-2017 competition (Task 7 Detection and Interpretation of English Puns), considering effects of using supervised learning models and word statistics.

Keywords: lexical ambiguity, pun, computational humor, thesaurus

РАСПОЗНАВАНИЕ НАМЕРЕННОЙ ЛЕКСИЧЕСКОЙ НЕОДНОЗНАЧНОСТИ В АНГЛИЙСКИХ КАЛАМБУРАХ

Михалькова Е. В. (e.v.mikhalkova@utmn.ru),
Карякин Ю. Е. (y.e.karyakin@utmn.ru)

ФГАОУ ВО «Тюменский государственный университет», Тюмень, Россия

1. Concerning puns

Computational humor is a branch of computational linguistics, which developed fast in the 1990s. Its two main goals are interpretation and generation of all kinds of humor.[1] Recently we noticed a new rise of attention to this research area, especially

[1] In [Mikhalkova 2010] we gave a brief account of main trends in computational humor up to 2010.

concerning analysis of short genres like tweets [Davidov et al. 2010; Reyes et al. 2013; Castro et al. 2016]. Furthermore, a number of tasks at SemEval-2017 (an annual event, organized by the Association for Computational Linguistics) was about analyzing short funny utterances, like humorous tweets (Task 6: #HashtagWars: Learning a Sense of Humor) and puns (Task 7: Detection and Interpretation of English Puns). The following article is an extended review of the algorithm that we used for pun recognition in SemEval, Task 7.

In [Miller et al. 2015], Tristan Miller and Iryna Gurevych give a comprehensive account of what has already been done in automatic recognition of puns. They note that the study of puns mainly focused around phonological and syntactic, rather than semantic interpretation. At present, the problem of intentional lexical ambiguity is viewed more as a WSD-task, solving which is not only helpful in detecting humor, but can also provide new algorithms of sense evaluation for other NLP-systems.

The following terminology is basic in our research of puns. **A pun** is a) a short humorous genre, where a word or phrase is used intentionally in two meanings, b) a means of expression, the essence of which is to use a word or phrase so that in the given context the word or phrase can be understood in two meanings simultaneously. **A target word** is a word, used in a pun in two meanings. **A homographic pun** is a pun that “exploits distinct meanings of the same written word” [Miller et al. 2015] (these can be meanings of a polysemantic word, or homonyms, including homonymic word forms). **A heterographic pun** is a pun, in which the target word resembles another word or phrase in spelling; we will call the latter **the second target word**. More data on classification of puns and their elaborated examples can be found in [Hempelmann 2004].

(1) *I used to be a banker, but I lost interest.*

Ex. 1 (the Banker joke) is a homographic pun; “interest” is the target word.

(2) *When the church bought gas for their annual barbecue, proceeds went from the sacred to the propane.*

Ex. 2 (the Church joke) is a heterographic pun; “propane” is the target word, “profane” is the second target word.

Our model of automatic pun detection is based on the following premise: in a pun, there are two groups of words and their meanings that indicate the two meanings, in which the target word or phrase is used. These groups overlap, i.e. contain the same words, used in different meanings.

In Ex. 1, words and collocations “banker”, “lost interest” point at the professional status of the narrator and his/her career failure. At the same time, “used to”, “lost interest” tell a story of losing emotional attachment to the profession: the narrator lost curiosity. We propose an algorithm of homographic pun recognition that discovers these two groups of words and collocations, based on common semes[2], which words in these groups share. When the groups are found, in homographic puns, the next step is to state where these groups overlap, and choose which word is the target word. In case of heterographic puns, the algorithm looks for the word or phrase, which

[2] We understand a seme as a minimal bit of meaning.

is used in one group and *not* used in the other. The last step in the analysis of heterographic puns is to calculate the second target word[3].

2. Mining semantic fields

We will call a semantic field a group of words and collocations[4] that share a common seme. We hold by the opinion that the following reciprocal dependency between a word and a seme is true: in a bunch of words, the more abstract a seme is, the more words share it, and vice versa the more there are words that share a seme, the more abstract the seme is. This type of relations between lexical items can be found in taxonomies, like WordNet [Fellbaum 1998] and Roget's Thesaurus [Roget 2006] (further referred to as Thesaurus). Applying such dictionaries to get the common groups of words in a pun is, therefore, the task of finding two most general hypernyms in WordNet, or two relevant Classes among the six Classes in Thesaurus. We chose Thesaurus, as its structure is not more than five levels deep, Classes labels are not lemmas themselves, but arbitrary names (we used numbers instead), and it allows parsing on a certain level and insert corrections. After some experimentation, instead of Classes, we chose to search for relevant Sections, which are 34 subdivisions[5] of the six Classes.

(3) *I wasn't originally going to get a brain transplant, but then I changed my mind.*

Beside its structure, Thesaurus contains many collocations; these are not only multiword units, but also aphorisms, proverbs, etc. The collocations have their own position in Thesaurus, different from the words, which compose them. Preliminary research showed the importance of collocations, in which target words appear. Sometimes the whole pun stands on rethinking a stable union of words, like in Ex. 3, "to change one's mind" becomes "to change one's brain". Therefore, when the semantic fields in a pun are discovered, it is sometimes crucial that the algorithm also analyzes collocations. In the current implementation, our program extracts collocations, based on their morphological composition. The following patterns are used: (verb+particle), (verb+(determiner/pronoun)[6]+noun+((conjunction/preposition)+noun)), (verb+adverb), (adverb+participle), (adjective+noun), (noun+(conjunction/preposition)+noun). Whenever a pattern appears in a sentence, the program checks for a collocation in Thesaurus and harvests its meaning.

The algorithm collects Section numbers for every word and collocation and removes duplicates (in Thesaurus homonyms proper can be assigned to different subdivisions in the same or different sections), excluding stop words like "to", "a" etc.[7] Table 1 illustrates to what sections words in Ex. 1 belong.

[3] In the current article, we will not consider algorithms we used to assign a Wordnet definition to a target word. This issue will be addressed in further research.

[4] By collocations, we mean "expressions of multiple words which commonly co-occur" [Bird et al. 2009].

[5] Sections are not always immediate subdivisions of a Class. Some Sections are grouped in Divisions.

[6] The inside parentheses show that this part of the phrase may be missing; a slash stands for "or".

[7] Stopwords are excluded from semantic analysis, but not from collocation extraction.

Table 1. Semantic fields in the Banker joke

Word	Section No., Section name in Thesaurus
I	—
use	24, Volition In General 30, Possessive Relations
to	—
be	0, Existence 19, Results Of Reasoning
a	—
banker	31, Affections In General 30, Possessive Relations
but	—
lose	21, Nature Of Ideas Communicated 26, Results Of Voluntary Action 30, Possessive Relations 19, Results Of Reasoning
interest	30, Possessive Relations 25, Antagonism 24, Volition In General 7, Causation 31, Affections In General 16, Precursory Conditions And Operations 1, Relation

Then the semantic vector of a pun is calculated. Every pun is a vector in a 34 dimensional space:

$$p_i = p_i(s_{1i}, s_{2i}, \dots, s_{34i}) \qquad (1)$$

The value of every element s_{ki} equals the number of words in a pun, which belong to a Section S_k:

$$S_{ki} = \sum_{j=1}^{l_i} \{1 | w_{ji} \in S_k\}, \qquad k = 1,2,\dots,34, \qquad i = 1,2,3\dots \qquad (2)$$

For example, the semantic vector of the Banker joke looks as follows:

$$p_{Banker} = \{1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,2,0,1,0,0,2,1,1,0,0,0,4,2,0,0\}$$

To test the algorithm, we, first, collected 2,480 puns from different Internet resources and, second, built a corpus of 2,480 random sentences of length 5 to 25 words from different NLTK [Bird et al. 2009] corpora[8] plus several hundred aphorisms and proverbs from different Internet sites. We shuffled and split the sentences into two equal groups, the first two forming a training set and the other two a test set. The classification was conducted using different Scikit-learn [Pedregosa et al. 2011] algorithms. In all the tests, the Scikit-learn algorithm of SVM with the Radial Basis Function (RBF) kernel produced the highest average F-measure results ($\overline{f} = \frac{f_{puns}+f_{random}}{2}$). In addition, its results are smoother, comparing the difference between precision and recall (which leads to the

[8] Mainly Reuters, Web corpus and Gutenberg.

highest F-measure scores) within the two classes (puns and random sentences), and between the classes (average scores).

Table 2 illustrates results of different Scikit-learn algorithms, applied in classification of puns against two selections of random sentences: the first one (**Mixed styles**) is a mixture of Brown, Reuters and Web NLTK corpora, the second one (**Belles lettres**) contains sentences from Gutenberg (also NLTK), some proverbs and aphorisms. As the learning algorithms are widely used in NLP, we provide only their names. Their full description can be found in Scikit-learn documentation [Pedregosa et al. 2011]. The results given are a mean of five tests.

Table 2. Tests for pun recognition

Method	Precision		Recall		F-measure	
	Pun	Not pun	Pun	Not pun	Pun	Not pun
Mixed styles						
SVM with linear kernel	0.68	0.66	0.63	**0.71**	0.66	0.68
SVM with Radial Basis Function (RBF) kernel	**0.70**	**0.68**	**0.67**	**0.71**	**0.68**	**0.70**
Belles lettres						
SVM with linear kernel	**0.75**	0.69	0.65	**0.78**	0.69	0.73
SVM with Radial Basis Function (RBF) kernel	0.74	**0.73**	**0.72**	0.74	**0.73**	**0.74**
Logistic Regression	0.74	0.70	0.67	0.76	0.70	0.73

All the algorithms worked better in comparison of puns to literature, proverbs and aphorisms, the performance increasing by several percent. Moreover, within each class, SVM with the RBF kernel produced most of the highest results. The reason for this is most likely caused by the topicality issue: compared to random sentences many puns tackle similar issues, and even use recurring realias (for example, John Deere, appearing in 7 different puns). To see how big its influence is, we changed vectors, sorting numbers in them in a decreasing order, and retested the algorithms. First, the whole vector was sorted out (**First sorting** in Table 3); second, the initial vector was split into four parts of sizes 8, 8, 8, 10, and sorting was done within each part[9] (**Second sorting** in Table 3).

Table 3. Tests for pun recognition: reduced topicality

Method	Precision		Recall		F-measure	
	Pun	Not pun	Pun	Not pun	Pun	Not pun
First sorting						

[9] The vector of the Banker joke now looks as follows: {1, 1, 1, 0, 0, 0, 0, 0 | 0, 0, 0, 0, 0, 0, 0, 0 | 2, 1, 1, 0, 0, 0, 0, 0 | 4, 2, 2, 1, 1, 0, 0, 0, 0, 0}.

Method	Precision		Recall		F-measure	
	Pun	Not pun	Pun	Not pun	Pun	Not pun
SVM with linear kernel	0.57	0.56	0.56	0.57	0.56	0.57
SVM with Radial Basis Function (RBF) kernel	**0.58**	**0.58**	**0.57**	**0.59**	**0.58**	**0.58**
Second sorting						
SVM with linear kernel	**0.70**	0.64	0.57	**0.75**	0.63	**0.69**
SVM with Radial Basis Function (RBF) kernel	0.68	**0.66**	**0.64**	0.70	**0.66**	0.68

After sorting, the difference between RBF and linear kernel becomes very low, but RBF is still (and inexplicably) more successful. The first sorting results in 58% success on average, when chance classification would produce a 50% result. The difference in 8% shows the purely structural potential of the algorithm, which, probably, rises from the curve of the semantic vector (differences among the most representative semantic fields, the "tail" of less representative fields, etc.). The partitioned sorting increases results by 10%, although the vector splits into four parts only. Splitting the vector into three parts results in 1% rise (not reflected in the table), which can be a feature of this particular dataset, or a more general trend, but this hypothesis requires more research.

The decrease in results shows that topicality is of much influence in pun recognition, although by definition a pun is not sense biased. This brings us to the idea that topicality is influential in puns as a *humorous* genre. Judging from the definition of a pun, as a means of expression, it can occur in any semantic context. However, puns, as a humorous genre, must inherit topical trends of humor. Some theories discuss existence of such trends. For example, R. Mihalcea and C. Strapparava write of "weak" human moments and targeting "professional communities that are often associated with amusing situations, such as lawyers, programmers, policemen like" in one-liners [Mihalcea et al. 2006: 139]. In [Mikhalkova 2009], we studied topical trends of physical and mental disorders, disorderly behavior, courtship, eating habits, and some other in comic TV-shows; etc.

3. Hitting the target word

We suggest that in a homographic pun the target word is a word that immediately belongs to two semantic fields; in a heterographic pun the target word belongs to at least one discovered semantic field and does not belong to the other. However, in reality, words in a sentence tend to belong to too many fields, and they create noise in the search. To reduce influence of noisy fields in the model, we included such a non-semantic feature as the tendency of the target word to occur closer to the end of a pun [Miller et al. 2015].

A-group (W_A) and B-group (W_B) are groups of words in a pun, which belong to the two semantic fields, sharing the target word. A-group attracts the maximum number of words in a pun:

$$S_{Ai} = \max_{k} S_{ki}, \qquad k = 1,2,\dots,34 \tag{3}$$

In the Banker joke s_{Ai} = 4, A = 30 (Possessive Relations); words that belong to this group are 'use', 'lose', 'banker', 'interest'.

B-group is the second largest group in a pun:

$$S_{Bi} = \max_{k}(S_{ki} \backslash S_{Ai}), \qquad k = 1,2,\dots,34 \tag{4}$$

In the Banker joke $s_{Bi} = 2$. There are three groups of words that have two words in them: B_1 = 19, Results Of Reasoning: "be", "lose"; $B_2 = 24$, Volition In General: "use", "interest'; $B_3 = 31$, Affections In General: "banker", "interest". Ideally there should be a group of about three words and collocations, describing a person's inner state ("used to be", "lose", "interest"), and two words ("lose", "interest") in W_A are a target *phrase*. However, due to the shortage of data about collocations in dictionaries and limitations of the collocation extraction algorithm, W_B divides into several smaller groups. Consequently, to find the target word, we appeal to other word features. In testing the system on homographic puns, we relied on polysemantic character of words. If in a joke, there are more than one value of B, W_B candidates merge into one, with duplicates removed, and every word in W_B becomes the target word candidate: $(c \in W_B)$. In Ex. 1 W_B is a list of "be", "lose", "use", "interest", "banker"; $B = \{19,24,31\}$.

Based on the definition of the target word in a homographic pun, words from W_B, that are also found in W_A, should have a privilege. Therefore, the first value (v_α), each word gets, is the output of the Boolean function:

$$v_\alpha(c) = g(c, W_A, W_B) = \begin{cases} 2, & if (c \in W_A) \wedge (c \in W_B) \\ 1, & if (c \notin W_A) \wedge (c \in W_B) \end{cases} \tag{5}$$

The second value (v_β) is the absolute frequency of a word in $W_{B_1} \cup W_{B_2} \cup W_{B_3}$ (the union of B_1, B_2, etc., including duplicates:

$$v_\beta(c) = f_c(W_{B_1} \cup W_{B_2} \cup W_{B_3})$$

Together values v_α and v_β compose a group of sense criteria. In case of target word candidates, we multiply them and choose the word with the maximum rate:

$$z_1(W_B) = \left\{ c \mid \max_{c}(v_\alpha \times v_\beta) \right\} \tag{6}$$

The reasons for using plain multiplication in the objective function (6) lie in our treatment of puns properties. In the algorithm, they are maximization criteria: the more properties the sentence has and the more represented they are, the more likely the sentence is a pun. Grounded by maximization criteria, the word with the maximum rate is, therefore, the best candidate for the target word. In case of a tie, the algorithm picks up a random candidate.

Another way to locate the target word is to rely on its position in a pun v_γ: the closer it is to the end, the bigger this value is. If the word occurs several times, the algorithm counts the average of sums of position numbers. The output is again the word with the maximum value.

Values of the Banker joke are illustrated in Table 4.

Table 4. Values of the Banker joke

Word form	v_α	v_β	$z_1(W_B)$	v_γ
be	1	1	1	4
lose	2	1	2	9
use	2	1	2	2
interest	2	2	**4**	**10**
banker	2	1	2	6

As for heterographic puns, the target word is missing in W_B (the reader has to guess the word or phrase, homonymous to the target word). Accordingly, we rely on the completeness of the union of W_A and W_B: among the candidates for W_B (second largest groups) such groups are relevant, that form the longest list with W_A (duplicates removed). In Ex. 2 (the Church joke) W_A = {'go', 'gas', 'annual', 'barbecue', 'propane'}, and two groups form the largest union with it: W_B = {'buy', 'proceeds'} + {'sacred', 'church'}. Every word in W_A and W_B can be the target word.

Due to sorting conditions, frequencies are of no value here; therefore, the method uses only the value of position in the sentence v_γ. The function output is:

$$z_2(W_A W_B) = \left\{c \mid \max_c(v_\gamma)\right\} \tag{7}$$

Values of the Church joke are illustrated in Table 5.

Table 5. Values of the Church joke

Word form	v_γ
propane	**18**
annual	8
gas	5
sacred	15
church	3
barbecue	9
go	12
proceeds	11
buy	4

We tested the suggested algorithms on SemEval Gold data. Table 6 illustrates percentage of correct guesses within a pun (True Positive results).

SemEval organizers suggested their baselines for this task: selecting 1) a random word, 2) the last word in a pun, 3) the word with the biggest number of senses (the most polysemantic word) [Miller et al. 2017]. We also include their results in the table.

Table 6. Test results of target word analysis

	Homographic puns	**Heterographic puns**
Sense-based method, $z_1(W_B)$	0.2373	—
Last word method, v_γ	**0.5145**	0.3879
SemEval random	0.0846	0.0839
SemEval last word	0.4704	**0.5704**
SemEval polysemantic word	0.1798	0.0110

Concerning homographic puns, the Last word method appears to be more effective, compared to SemEval last word, probably, due to the lack of filter for content words. At the same time, our Sense-based method is more effective than SemEval polysemantic word.

The Last word solution for heterographic puns turns out to be 18% less effective, than SemEval baseline (0.39 and 0.57, correspondingly). Testing heterographic puns with the algorithm for homographic puns brought even lower results. The reason for it, probably, lies in the method itself, that lacks the sense criterion about the target word present in one semantic group and absent in the other. This will be the only significant difference from the solution for homographic puns, beside a special treatment of W_B.

4. Results of SemEval-2017

Tables 7 and 8 reflect the top-scoring results of SemEval-2017, Task 7: Detection and Interpretation of English Puns, given in [Miller et al. 2017], and results of the own system PunFields (at competition and currently). Table 7 shows results for the class of puns. Table 8 shows Precision.

Table 7. SemEval pun classification

	Homographic puns			**Heterographic puns**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Duluth	0.7832	0.8724	0.8254	0.7399	0.8662	0.7981
Idiom Savant	—	—	—	**0.8704**	0.8190	**0.8439**
JU_CSE_NLP	0.7251	0.9079	0.8063	0.7367	**0.9402**	0.8261
N-Hance	0.7553	**0.9334**	**0.8350**	0.7725	0.9300	**0.8440**
PunFields	**0.8019**	0.7785	0.7900	0.7585	0.6326	0.6898
PunFields, current result[10]	0.75	0.72	0.73	0.75	0.72	0.73

[10] Currently, we do not test homographic and heterographic puns separately.

Table 8. SemEval pun location

	Homographic puns	Heterographic puns
Idiom Savant	**0.6636**	0.6845
U-Waterloo	0.6526	**0.7973**
N-Hance	0.4269	0.6592
PunFields	0.3279	0.3501
PunFields, current result	0.5145	0.3879

PunFields participated in SemEval-2017, Task 7 in a slightly different form. In pun classification (Paragraph 2), together with the collection of 2,480 puns, it used the Belles lettres corpus as a training set. In the present research the training set is twice smaller. Hence, the difference in results. The current result for pun location (Paragraph 3) is more valid, due to rethinking of sorting criteria and elimination of minor coding errors.

Generally, PunFields was most successful in pun classification, which can be due to advantages of supervised learning. Although there were other less successful systems, also using supervised learning algorithms.

SemEval winning systems in pun classification did not have much in common. Duluth used several WordNet customizations, some designed by its author T. Pedersen [Pedersen et al. 2009]. When these customizations disagree, the sentence is classified as a pun. IdiomSavant is a combination of different methods, including word2vec. JU_CSE_NLP is a supervised learning classifier, combining a hidden Markov model and a cyclic dependency network. N-Hance is a heuristic, making use of Pointwise Mutual Information, calculated for a list of word pairs[11]: the algorithm sorts out sentences, where the highest PMI is distinctively higher than its lower neighbor.

Concerning pun location, there were two systems that outperformed SemEval baseline by nearly 20%: Idiom Savant, described above, and UWaterloo. UWaterloo has 11 criteria to calculate the target word (word frequency, part-of-speech context, etc.), but again focuses on the second half of a pun. The system description papers have not been released so far, and it is hard to work out the main factor in the success of these two systems.

It is of interest that the simple approach, suggested by N-Hance, turned out to be so effective. Unlike other winners, it is not a supervised learning classifier or a combination of methods, some of which can be supple to tuning into a dataset. However, it was not as effective in pun location as in classification, and again the search was done among second elements of the pair with the highest PMI score (the end of the sentence criterion).

Corollaries

We consider that the results of the present research allow us to state the following: the hypothesis about two semantic fields, underlying in every pun, is relatively true and objective; Roget's Thesaurus is a credible source in automatic semantic analysis; the semantic nature of puns (and other kinds of metaphorical language

[11] PMI measure was calculated on the basis of a Wikipedia corpus.

issues) can be subject to exact sciences. The suggested algorithm of pun detection and interpretation is fairly effective, but requires improvement. We tend to think that PunFields has advantageous prospects in customizing it to WordNet.

The research also objectivizes some fundamental issues in understanding humor. One of them is topicality bind. There have been many suppositions and separately collected facts that humor is not universal, and that it thrives on some topics better than on other. Our pun classifier supports this trend.

In addition, we would like to stress the importance of phrases in creation of lexical ambiguity. Even in puns, where only one word is obviously ambiguous, its neighbors can have shades of other possible meanings. In the Banker joke, "lose" in collocation with "interest" can be antonym to "win, earn" in connection with "money, benefit", and to "get, gain" in connection with "curiosity".

Concerning location of the target word in a pun, competition results show that the structural "closer to the end" criterion is of great importance and is hard to beat even as the baseline. This issue has also been discussed in theories of humor: punchlines and target words do tend to occur at the end of an utterance.

SemEval competition included one more task: assigning a WordNet definition to the target word. This task appeared to be the most difficult, and very few systems beat the baseline results, which also leaves us grounds for further work.

References

1. *Bird S., Klein E., Loper E.* (2009), Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc.
2. *Castro S., Cubero M., Garat D., Moncecchi G.* (2016, November), Is This a Joke? Detecting Humor in Spanish Tweets, Ibero-American Conference on Artificial Intelligence, Springer International Publishing, pp. 139–150.
3. *Davidov D., Oren Tsur, Rappoport A.* (2010), Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon, Proceedings of the Fourteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, pp.107–116.
4. *Fellbaum C.* (1998, ed.), WordNet: An Electronic Lexical Database, Cambridge, MA, MIT Press.
5. *Hempelmann C.* (2004), Script Opposition and Logical Mechanism in Punning, Humor, 17–4, pp. 381–392.
6. *Mihalcea R., Strapparava C.* (2006), Learning to laugh (automatically): Computational models for humor recognition, Computational Intelligence, 22(2), pp. 126–142.
7. *Mikhalkova E. V.* (2009), Pragmatics and Semantics of Invective in Mass Media Discourse (Based on Russian and American Comic TV-Shows) [Pragmatika i semantika invektivy v massmedijnom diskurse (na materiale russkih i amerikanskih komedijnyh teleshou)], Abstract of Candidate's Degree Thesis, Tyumen.
8. *Mikhalkova E. V.* (2010, September), A Theory of Invective Names: Possibilities in Formalizing Humorous Texts [Koncepcija invektivnyh imen: vozmozhnosti primenenija dlja formalizacii smysla komicheskih tekstov], Proceedings

of KII-2010: The Twelfth National Conference on Artificial Intelligence [KII-2010: Dvenadcataja nacional’naja konferencija po iskusstvennomu intellektu s mezhdunarodnym uchastiem], Vol. 1, pp. 201–208.

9. *Miller T., Gurevych I.* (2015), Automatic Disambiguation of English Puns, The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: Proceedings of the Conference (ACL–IJCNLP), Vol. 1, Stroudsburg, PA: Association for Computational Linguistics, July 2015, pp. 719–729.
10. *Miller T., Hempelmann C., Gurevych I.* (2017), SemEval-2017 Task 7: Detection and Interpretation of English Puns, Draft.
11. *Pedersen T., Kolhatkar V.* (2009), Word-Net::SenseRelate::AllWords—a broad coverage word sense tagger that maximizes semantic relatedness, Proceedings of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies 2009 Conference, Boulder, CO, pp. 17–20.
12. *Pedregosa F. et al.* (2011), Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825–2830.
13. *Reyes A., Rosso P., Veale T.* (2013), A Multidimensional Approach for Detecting Irony in Twitter, Language Resources and Evaluation, 47.1, pp. 239–268.
14. *Roget P. M.* (2004), Roget’s Thesaurus of English Words and Phrases, Project Gutenberg.

DISTRIBUTIONAL SEMANTIC FEATURES IN RUSSIAN VERBAL METAPHOR IDENTIFICATION[1]

Panicheva P. V. (ppolin86@gmail.com),
BadryzlovaYu. G. (yuliya.badryzlova@gmail.com)

St. Petersburg State University, Saint Petersburg, Russia;
National Research University Higher School of Economics (HSE), Moscow, Russia

Our experiment is aimed at evaluating the performance of distributional semantic features in metaphor identification in Russian raw text. We apply two types of distributional features representing similarity between the metaphoric/literal verb and its syntactic or linear context. Our approach is evaluated on a dataset of nine Russian verb context, which is made available to the community. The results show that both sets of similarity features are useful for metaphor identification, and do not replicate each other, as their combination systematically improves the performance for individual verb sense classification, reaching state-of-the-art results for verbal metaphor identification. A combined verb classification demonstrates that the suggested features effectively generalize over metaphoric usage in different verbs, shows that linear coherence features perform as well as the combined feature approach. By analyzing the errors we conclude that syntactic parsing quality is still modest for raw-text metaphor identification in Russian, and discuss properties of semantic models required for high performance.

Keywords: Metaphor identification, Russian language, distributional semantics, contextual anomaly

[1] The contribution to this study by Polina Panicheva is supported by RFBR grant № 16-06-00529 "Development of a linguistic toolkit for semantic analysis of Russian text corpora by statistical techniques". The work of Yulia Badryzlova is an output of a research project implemented as part of the Basic Research Program at the National Research University Higher School of Economics (HSE).

ДИСТРИБУТИВНАЯ СЕМАНТИКА В АВТОМАТИЧЕСКОМ ВЫЯВЛЕНИИ ГЛАГОЛЬНОЙ МЕТАФОРЫ В РУССКОМ ЯЗЫКЕ

Паничева П. В. (ppolin86@gmail.com),
Бадрызлова Ю. Г. (yuliya.badryzlova@gmail.com)

Санкт-Петербургский государственный университет,
Санкт-Петербург, Россия;
Национальный исследовательский университет
«Высшая школа экономики», Москва, Россия

1. Introduction

Metaphor processing has in recent years gained high popularity, both for practical and fundamental reasons. Not only is it indispensable to account for metaphor in various language processing tasks, but it is also commonly accepted that metaphor is a pervasive process in human language and thought (Lakoff and Johnson, 2008), with numerous effects in cognitive disciplines. There have been various effective approaches to automatic identification of linguistic metaphor, mostly relying on a consideration that metaphor is a violation of contextual selectional restrictions. Recent approaches to metaphor identification escape the subjectivity and sparseness of hand-coded semantic resources by applying distributional semantic models.

We evaluate word-embeddings distributional semantic features in the task of metaphor identification on a raw text dataset of Russian verbs. Our goal is to assess a range of distributional features in a real-world text processing task. Syntax-based features presented in previous work are extended and combined with linear contextual anomaly detection techniques. While preserving the original selectional restriction violation view on metaphor, in practice we also position it in a range of phenomena characterized by contextual anomaly. Our features reach state-of-the-art performance, proving that distributional semantic features are an important technique in metaphor detection.

2. Related Work

2.1. Word Meaning Representation in Metaphor Identification

Most of the work on automatic metaphor identification is aimed at uncovering specific relations between metaphorically used words and their context. The idea is formulated in its strict version in the selectional restriction approach to metaphor (Mason, 2004; Shutova, 2010). Methods of analyzing the metaphor-context relations make use of two main information sources: symbolic and numeric. Symbolic approaches apply hand-coded data from external resources such as WordNet, FrameNet, MetaBank, as features representing word meaning (Martin, 1994; Mason, 2004; Peters and Peters, 2000; Gedigian et al., 2006; Krishnakumaran and Zhu, 2007; Shutova, 2010; Klebanov et al., 2016). Numeric approaches usually involve assessing word meaning on numeric scales representing different aspects of meaning, of which particularly useful are concreteness, imageability, animateness (Turney et al., 2011; Gandy et al., 2013; Tsvetkov et al., 2014).

There has been some important work on metaphor identification in Russian, reaching high performance of up to 0.77–0.85 F1; however, most of it has been focused on large hand-coded databases or dictionaries (Ovchinnikova et al., 2014; Mohler et al., 2014; Tsvetkov et al. 2014).

2.2. Distributional Features in Metaphor Identification

Distributional semantic modeling allows to evaluate relations between word meanings in their entirety, automatically build conceptual domain knowledge, at the same time overcoming subjectivity and data sparseness characteristic of hand-coded resources. Authors of (Heintz et al., 2013) have used LDA topic modeling to automatically induce source domains for metaphors in political domain. In (Strzalkowski et al., 2013) candidates for metaphoric usage are identified by excluding the 'topic chains' forming the main topical structure of a text. Klebanov et al. (2016) evaluate corpus-based distributional semantic features against hand-coded resources. Shutova et al. (2016) have applied word-embeddings to measuring metaphoricity in word pairs. They classified verb-subject, verb-object and noun-attribute word pairs as metaphorical or literal, based on semantic relatedness measures between the paired words, achieving best performance among linguistic methods (F1 = 0.71–0.76).

3. Experiment

3.1. Metaphor Annotation

The metaphoric occurrences of verbs were defined in accordance with the Vrije Universiteit Metaphor Identification Procedure (MIPVU) (Steen et al., 2010). MIPVU defines the three major types of metaphor-related words: Indirect Metaphor, Possible Personification, and Direct Metaphor.

Indirect Metaphor is attested when the contextual meaning of a word is not basic. The basic meaning of a word "is defined as a more concrete, specific, and human-oriented sense in contemporary language use" (ibid., p. 35).

For example, the basic meaning of 'взорвать' — to blow up — is 'to explode smth, to destroy smth by an explosion'. The non-basic meanings are:

1. 'to make a sensational impression on smb, to astonish smb';
2. 'to outrage, to scandalize smb';
3. 'to trigger a sudden and drastic change' (Evgenyeva, 1981).

Thus, Indirect Metaphor in MIPVU covers the cases of conventional lexicalized metaphor, e.g.:

(1) *‘Обещания премьера ... взорвали блогосферу.’ — Prime Minister’s promises have ... exploded the blogosphere.*

Direct Metaphor is attested when lexical units are used in their direct sense, but they are “incongruous” with the ‘overall referential and/or topical framework’ of their context, so their use can be ‘potentially explained by cross-domain mapping’ (ibid., p. 38):

(2) *‘Мы ... тебе особую, яркую судьбу выкраивали.’ — It was an extraordinary and outstanding destiny that … we were cutting out for you.*

Besides, Direct Metaphor in MIPVU includes the cases of simile; they are termed as ‘flagged’ metaphor-related words:

(3) *Словно бы я желал выкроить из бумаги твое пение! — As if I wanted to cut your singing out of paper!*

MIPVU’s Possible Personification covers the two types of occurrences: a) when an argument of a word is expressed metonymically and b) when an argument violates the selectional preference of a word for animate arguments:

(4) *Департамент их [деньги] пилит. — The department is sawing (=embezzling) the money.*

(5) *Прошлогодняя гигантская лавина … аккуратно «причесала» ... все оставшиеся кустики и деревья… — Last year’s gigantic avalanche … thoroughly ‘combed’ … the remaining shrubs and trees …*

Indirect and Direct Metaphor, and Possible Personification together constitute a single class of metaphor-related words forming a binary opposition with non-metaphoric occurrences. Our experimental corpus contains Indirect Metaphor and Possible Metonymic Personification (including examples (1), (4)), while Direct Metaphor (examples 2,3) and Metaphoric Personification (5) were excluded from the analysis.

Although graphic cues (such as quotation marks) are not distinguished in MIPVU as signals of metaphor, they definitely serve as supplementary factors that enable annotators to identify metaphoric occurrences.

MIPVU, designed to provide an explicit and standardized protocol for metaphor identification, has shown reliable levels of inter-annotator agreement: Fleiss’ kappa > 0.79 in English and Dutch (Steen et al., 2010), and 0.9 in Russian texts (Badryzlova et al., 2013).

3.2. Dataset

We experimented with nine Russian polysemous transitive verbs: *бомбардировать* — ’bombardirovat’ — to bombard, *очертить* — ’ochertit’ — to outline, *пилить* — ’pilit’ — to saw, *распылять* — ’raspylyat’ — to diffuse, *разбавлять* — ’razbavlyat’ — to dilute, *выкраивать* — ’vykraivat’ — to cut out, *взорвать* — ’vzorvat’ — to blow up, *взвесить* — ’vzvesit’ — to weigh, and *зажигать* — ’zazhigat’ — to ignite.

The dataset contained 990 full sentences (roughly about 100 sentences per verb), with an approximately equal number of metaphoric and non-metaphoric occurrences for each target verb.

The sentences were randomly sampled from RuTenTen, a Russian web corpus accessed via the Sketch Engine tool[2]. The sentences were manually annotated as metaphoric or non-metaphoric by a native Russian speaker who is a trained linguist[3].

Table 1 shows the number of sentences for each verb, and the majority class occurrence.

Table 1: Dataset summary

Verb	Sentences	Majority, %
бомбардировать — bombard	120	55
очертить — outline	99	53
пилить — saw	106	52
распылять — diffuse	112	50
разбавлять — dilute	115	51
выкраивать — cut out	110	53
взорвать — blow up	96	50
взвесить — weigh	133	51
зажигать — ignite	99	53
All	990	50

3.3. The task of distributional metaphor classification

The goal of the work is to evaluate the performance of distributional semantic measures in Russian verbal metaphor identification. The task is to distinguish metaphoric from literal usage of a verb in raw, full sentence context. To achieve this, we apply a number of distributional measures characterizing the semantic relations between the verb and the context.

Our setting is crucially different from the tasks described by Shutova et al. (2016) and Tsvetkov et al. (2014) in that we process raw contexts of verbs, and not hand-picked syntactically related word pairs. Another complication of the raw-text task is that a verb paired with it's syntactic dependency cannot always be unambiguously resolved in terms of metaphoricity/literacy, unlike the unequivocally annotated word-pairs in previous work. Consider example (6), where the direct object *круг — circle* does not resolve the metaphoricity/literacy ambiguity: (6a) is an ambiguous word pair and can only be resolved by using broader context, as illustrated by (6b, c):

(6a) *очертить круг — to outline a circle*
(6b) *очертить круг обязанностей — to outline a circle of responsibility*
(6c) *очертить круг палкой на песке — to outline a circle on the ground with a stick*

[2] http://www.sketchengine.co.uk/

[3] The dataset is available for download at http://web-corpora.net/~badryzlova/VERB_DATASET/.

Our task is thus complicated by the following steps necessary for real-world metaphor identification in raw text:

- identifying related syntactic arguments;
- overcoming the absence or ambiguity of specific syntactic arguments, or the parser's failure to identify them, by using broader context features.

We apply the following distributional features:

- a set of syntax-based features presented by Shutova et al. (2016);
- an extension of the syntax-based features to include all the significant dependency types;
- linear topic coherence features proposed by Newman et al. (2010).

We are concerned with the question how well can distributional measures perform in the real-world task of classifying contextual metaphoricity by applying the available state-of-the-art morphosyntactic processing and semantic modeling in Russian.

To our knowledge, this is the first work on automatic metaphor identification in Russian raw text samples by applying distributional techniques without relying on large-scale hand-crafted resources. It is also the first attempt to apply a Russian word-embeddings model with various contextual and syntax-based distributional semantic measures to metaphor identification. Our evaluation data is made available to the community.

3.4. Distributional Semantic Features

Distributional semantic models: Our distributional features are based on word-embeddings semantic models. We evaluate two pre-trained models presented in (Kutuzov and Andreev, 2015)[4]:

- RNC: a model trained with the Russian National Corpus texts containing 107M tokens, dimensionality = 300.
- WikiRNC: a larger model trained with a combined RNC + Russian Wikipedia dump corpus of 280M tokens, dimensionality = 500.

Both models have been trained using CBOW algorithm and window size 2.

Semantic Similarity (Sim): Semantic similarity features are based on the consideration that metaphor is a Selectional Preference violation, which is effectively captured as semantic deviance between the metaphoric verb and its main arguments (Shutova et al., 2016). The assumption is that a verb used in a literal sense belongs to the same conceptual domain as its immediate arguments, whereas metaphoric verb usage implies arguments belonging to a different conceptual domain. The semantic similarity features involve cosine similarity values between the verb and its syntactic dependencies: предик — Subject (Subj), компл-1 — Direct object (Obj), компл-2 — Indirect object (Obj2), сочин — Coordinate (Coord), and обст — Circumstance (Circ). Syntactic dependencies are identified by MaltParser (Sharov and Nivre, 2011). Semantic similarity features are calculated as follows:

[4] Freely available for download at http://ling.go.mail.ru/dsm/en/about#models

$$\mathrm{Sim_{rel}} = \cos(\mathrm{verb}, \mathrm{w_{rel}}) \tag{1),}$$

where *rel* is the syntactic relation in {Subj, Obj, Obj2, Coord, Circ}, *verb* is the keyword verb, and w_{rel} is the syntactically dependency of the keyword verb in the current relation.

Linear Semantic Coherence (Coh): Semantic coherence features are evaluated as the topic coherence measure proposed by (Newman et al., 2010). The intuition is that a metaphoric verb is semantically deviant from its linear context window, affecting mean semantic similarity between the words in the window in a negative way, whereas a literally used verb belongs to the same conceptual domain as its context, making the contextual sub-space denser and adding to mean similarity (Herbelot and Kochmar, 2016). We apply 3 features representing linear semantic coherence:

$$\mathrm{Coh_{win}} = \frac{1}{\mathrm{length(Win)}} \sum\nolimits_{\mathrm{w_i,w_j \in Win}} \mathrm{Si\,m}\left(\mathrm{w_i};\mathrm{w_j}\right) \tag{2),}$$

$$\mathrm{CohV_{win}} = \frac{1}{\mathrm{length(Win)}} \sum\nolimits_{\mathrm{w_i,w_j \in Win, w_i \neq verb, w_j \neq verb}} \mathrm{Si\,m}\left(\mathrm{w_i};\mathrm{w_j}\right) \tag{3),}$$

$$\mathrm{CohDiff_{win}} = \mathrm{Coh_{win}} - \mathrm{CohV_{win}} \tag{4);}$$

where *Sim* is the cosine similarity measure in the distributional semantic space; and *Win* is the context window consisting of [-x; x] content words (nouns, verbs, adjectives or adverbs) around the keyword verb.

One the one hand, Coherence features reproduce the conceptual domain similarity information provided by the Sim values, without relying on the syntactic subtleties, including the syntactic parsing quality. On the other hand, being effectively applied to lexical error detection (Herbelot and Kochmar, 2016), Coherence features render the task of metaphor identification as a case of lexical anomaly detection in linear context.

The features are combined to perform binary metaphoricity/literalness classification using Support Vector Machine (SVM) classification with linear kernel[5]. We applied 3-fold cross-validation in experiments with individual verbs; in the combined dataset experiment we used 30-fold cross-validation in order to maintain comparable training/test set volume between all the experiments.

4. Metaphor Identification Results

Metaphor identification results based on Sim and Coh features are presented in Table 2, with the results for two distributional models following the format "RNC / WikiRNC result", and the highest performance for a single verb/combined verbs highlighted in bold.

[5] LinearSVC, as implemented in scikit-learn, (Pedregosa et al., 2011).

Table 2: Metaphor classification results, accuracy, in %

	Sim					
Verb	**Subj**	**Obj**	**Subj+Obj**	**All**	**Coh**	**Coh+ Sim**
бомбардировать — bombard	59 / 55	57 / 56	59 / 58	59 / 58	63 / 55	**68** / 58
очертить — outline	53 / 53	52 / 51	51 / 49	52 / 49	**58** / 57	55 / 53
пилить — saw	50 / 53	61 / 60	62 / 61	64 / 65	71 / 64	**74** / 71
распылять — diffuse	51 / 54	44 / **71**	43 / **71**	46 / **71**	46 / 54	49 / 68
разбавлять — dilute	56 / 54	68 / 70	70 / 73	72 / 77	90 / 83	90 / **93**
выкраивать — cut out	53 / 53	53 / 53	53 / 53	53 / 55	57 / 52	**58** / 53
взорвать — blow up	59 / 60	74 / 69	78 / 75	77 / 75	78 / 63	**81** / 75
взвесить — weigh	51 / 50	48 / 48	47 / 47	**56** / 50	54 / 50	**56 / 56**
зажигать — ignite	55 / 56	69 / 70	71 / 72	71 / 72	**80** / 78	76 / 77
All	53 / 52	57 / 57	59 / 59	62 / 62	**68** / 67	**68 / 68**

Evaluated **Sim** feature sets include Subj, Obj, their combination, and all five dependency features Subj, Obj, Obj2, Coord, Circ. **Coh** feature results are illustrated for context window = 2 (other window sizes have resulted in the same pattern with insignificant differences). In the joint **Coh+Sim** classification we apply **Coh** feature set combined with the set of all five **Sim** dependency features. The best classification result for all the verbs combined reaches **68% Accuracy** or **F1 = 0.71** for the Metaphor class.

5. Discussion

5.1. Distributional verb representation

The best results for different verbs range from just above the majority baseline (**53%**) to very accurate classification (**93%**). The combined classification performs reasonably high (**68%**), reaching the level of medium-hard individual verbs: this proves that the applied features can be generalized over different verbs, reflecting not only individual peculiarities in verb meanings, but common patterns of metaphoric/literal usage.

It is obvious that the results differ considerably between individual verbs, i.e. there are certain ’easy’ and ’difficult’ verbs for classification. A range of factors affect the individual performance:

- The better the verb is represented in the training corpus of the distributional model, the higher the resulting classification accuracy. Spearman’s r between verb frequency and the RNC-based joint **Sim+Coh** (window = 2) classification results reaches a moderate correlation value of 0.37. Moreover, the two most under-represented verbs occurring less than 200 times in RNC, *diffuse* and *cut out*, have the lowest RNC-based classification performance.
- Qualitative representation of the keyword verbs in the models affects the performance. The verb representation in the distributional models has been manually analyzed by evaluating the most similar verbs to the keyword verbs. The

models representing mostly the literal/technical sense of a verb give higher performance in metaphor classification, than those representing broader, metaphoric sense (cf. *diffuse*: *распыляться* — diffuse(refl), *абсорбировать* — absorb, *ожижать* — liquify, *перенасыщать* — supersaturate (WikiRNC), *уничтожать* — destroy, *сосредотачивать* — focus, *мобилизовать* — mobilize, *превращать* — transform, *разгонять* — scatter (RNC)).

- For most of the verbs, **Sim** and **Coh** features both contribute to the result, achieving higher performance when **Sim** and **Coh** are combined. However, as the strict regularities erode into broader common patterns in the all-verb classification, fine-grained syntax-based **Sim** features give no additional advantage over the linear **Coh** features.

5.2. Error Analysis

We have classified reasons for wrong classification in the case of **Sim** Subj+Obj features. The main error types are presented in Table 3.

Table 3: Error type statistics in Sim Subj+Obj classification results

Verb	Sparsity	Syntactic	Semantic
Total	40	55	5
Combined	48	44	8

Errors are attributed to data sparsity when either Subj or Obj was not represented in the data, or either of them was not expressed by a noun, making semantic similarity measurement impossible. Classification errors due to failures in syntactic parsing occurred when the parser incorrectly identified either Subj or Obj, or both of them. Semantic errors are the errors when both nominal Subj and Obj are present in the data and correctly parsed, but the wrong classification is due to a failure of the semantic model or the classifier algorithm.

The most common error in **Sim** Subj+Obj are errors due to inadequate syntactic parsing: they explain about 55% of overall errors, ranging from 39 to 73% across the verbs, or 44% in the combined verb classification. Syntactic errors are followed by errors due to data sparsity which comprise 40–48% of all errors in total.

6. Conclusions and Future Work

We have performed metaphor identification with raw contexts of nine Russian verbs by applying distributional semantic features based on similarity to the main arguments and linear coherence. Both feature sets have proven to be useful with considerable performance in the task well above the majority baseline, reaching **63–93% Accuracy** for individual verbs. More importantly, the suggested distributional features generalize reasonably well in a combined classification of nine verbs, reaching **68% Accuracy**, or F1 = 0.71. The result is comparable to the reported state-of-the-art results for Russian and reaches that reported for a similar resource-free setting

(F1 = 0.71 for verbal metaphor by Shutova et al. (2016)). We consider the performance reasonably high, taking into account the raw-text setting of our experiment and the absence of hand-coded dictionary resources among our features.

One of the main difficulties of the task is the sparsity of main verb arguments, including genuine absence of arguments in the sentence and failure of the syntactic parser to identify them; it is effectively overcome by adding linear semantic coherence features. A high quality distributional semantic representation of a keyword verb for metaphor identification should reflect primarily the literal meaning of the verb, i.e. the metaphoric sense should not be too conventionalized in the model.

Distributional semantic features are a useful source of information in metaphor identification. In future work aimed at high-quality performance of metaphor identification, other features should be added, including explicit selectional preference encoding and word-meaning aspects such as concreteness, imageability.

We have shown that algorithms capturing contextual anomaly both in terms of syntactic pairs and linear context are effective in describing metaphor. As the same algorithms have been applied to identifying non-compositional constructions (Bukia et al., 2016) and L2-learner errors (Herbelot and Kochmar, 2016), it is clear that these phenomena share similar distributional properties with metaphoric usage. However, it is important in future work to draw a line between the three different phenomena representing contextual anomaly.

References

1. *Badryzlova Yu. G., Isaeva E. V., Kerimov R. D., Shekhtman N. G.* (2013), Pravila Primeneniya Protsedury Lingvisticheskoy Identifikatsii Metafory (MIPVU) v Russkoyazychnom Korpuse: Lingvokognitivnyy Opyt (Utochneniya i Dopolneniya) [Applying the Rules of the Linguistic Metaphor Identification Procedure (MIPVU) to a Russian Corpus: Cognitive Linguistic Evidence (Revised and Extended).]. In Gumanitarnyy Vektor, issue 4 (36), Kemerovo, pp. 19–39.
2. *Bukia G. T., Protopopova E. V., Panicheva P. V., Mitrofanova O. A.* (2016), Estimating Syntagmatic Association Strength Using Distributional Word Representations. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2016", pp. 124–133.
3. *Evgenyeva, A. P.* (1981). Slovar' russkogo yazyka: V 4-kh t.[Dictionary of the Russian Language. In 4 vols.]. Moscow, Russkiy yazyk Publ, 1984.
4. *Gandy L., Allan N., Atallah M., Frieder O., Howard N., Kanareykin S., … Argamon S.* (2013), Automatic Identification of Conceptual Metaphors With Limited Knowledge. In Twenty-Seventh AAAI Conference on Artificial Intelligence.
5. *Gedigian M., Bryant J., Narayanan S., Ciric B.* (2006), Catching metaphors. In Proceedings of the Third Workshop on Scalable Natural Language Understanding, pp. 41–48. Association for Computational Linguistics.
6. *Havasi C., Speer R., Alonso J.* (2007), ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In Recent Advances in Natural Language Processing, pp. 27–29.

7. *Heintz I., Gabbard R., Srinivasan M., Barner D., Black D. S., Freedman M., Weischedel R.* (2013), Automatic extraction of linguistic metaphor with lda topic modeling. In Proceedings of the First Workshop on Metaphor in NLP, pp. 58–66.
8. *Herbelot A., Kochmar E.* (2016), 'Calling on the classical phone': a distributional model of adjective-noun errors in learners' English. The 26th International Conference on *Computational Linguistics, pp. 976–986*.
9. *Klebanov B. B., Leong C. W., Gutierrez E. D., Shutova E., Flor M.* (2016), Semantic classifications for detection of verb metaphors. In The 54th Annual Meeting of the Association for Computational Linguistics, p. 101.
10. *Krishnakumaran S., Zhu X.* (2007), Hunting elusive metaphors using lexical resources. In Proceedings of the Workshop on Computational Approaches to Figurative Language, pp. 13–20. Association for Computational Linguistics.
11. *Kutuzov, A., & Andreev, I.* (2015). Texts in, meaning out: neural language models in semantic similarity task for Russian. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2015", pp. 133–147.
12. *Lakoff G., Johnson M.* (2008), Metaphors we live by. University of Chicago press.
13. *Martin, J. H.* (1994), METABANK: A KNOWLEDGE-BASE OF METAPHORIC LANGUAGE CONVENTIONS. Computational Intelligence, 10(2), pp. 134–149.
14. *Mason Z. J.* (2004), CorMet: a computational, corpus-based conventional metaphor extraction system. Computational Linguistics, 30(1), pp. 23–44.
15. *Mohammad S. M., Shutova E., Turney P. D.* (2016), Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Joint Conference on Lexical and Computational Semantics*.
16. *Mohler M., Rink B., Bracewell D. B., Tomlinson M. T.* (2014), A Novel Distributional Approach to Multilingual Conceptual Metaphor Recognition. The 25th International Conference on Computational Linguistics, pp. 1752–1763.
17. *Newman D., Lau J. H., Grieser K., Baldwin T.* (2010), Automatic evaluation of topic coherence. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 100–108.
18. *Ovchinnikova E., Israel R., Wertheim S., Zaytsev V., Montazeri N., & Hobbs J.* (2014), Abductive inference for interpretation of metaphors. In Proceedings of the Second Workshop on Metaphor in NLP, pp. 33–41.
19. *Peters W., Peters I.* (2000), Lexicalised Systematic Polysemy in WordNet. Proceedings of the 2nd international conference on Language Resources and Evaluation.
20. *Sharoff S., Nivre J.* (2011), The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2011", pp. 591–604.
21. *Shutova E., Kiela D., & Maillard J.* (2016), Black holes and white rabbits: Metaphor identification with visual features. In Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 160–170.
22. *Shutova E.* (2010), Automatic metaphor interpretation as a paraphrasing task. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 1029–1037.

23. *Steen G. J., Dorst A. G., Herrmann J. B., Kaal A., Krennmayr T., Pasma T.* (2010), A method for linguistic metaphor identification: From MIP to MIPVU (Vol. 14). John Benjamins Publishing.
24. *Strzalkowski T., Broadwell G. A., Taylor S., Feldman L., Yamrom B., Shaikh S., … Elliott K.* (2013), Robust extraction of metaphors from novel data. In Proceedings of the First Workshop on Metaphor in NLP, pp. 67–76.
25. *Tsvetkov Y., Boytsov L., Gershman A., Nyberg E., Dyer C.* (2014), Metaphor detection with cross-lingual model transfer. The 52nd Annual Meeting of the Association for Computational Linguistics.
26. *Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., … Vanderplas J.* (2011), Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, pp. 2825–2830.
27. *Turney P. D., Neuman Y., Assaf D., & Cohen Y.* (2011), Literal and metaphorical sense identification through concrete and abstract context. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 680–690.

RHETORICAL STRUCTURE THEORY AS A FEATURE FOR DECEPTION DETECTION IN NEWS REPORTS IN THE RUSSIAN LANGUAGE

Pisarevskaya D. (dinabpr@gmail.com)

Institute for System Programming of the RAS, Moscow, Russia

The framework of the Rhetorical Structure Theory (RST) can be used to reveal the differences between structures of truthful and deceptive (fake) news. This approach was already used for English. In this paper it is applied to Russian. Corpus consists of 134 truthful and deceptive news stories in Russian. Texts annotations contain 33 relation categories. Three data sets of experimental data were created: with only rhetorical relation categories (frequencies), with rhetorical relation categories and bigrams of categories, with rhetorical relation categories and trigrams of categories. Support Vector Machines and Random Forest Classifier were used for text classification. The best results we got by using Support Vector Machines with linear kernel for the first data set (0.65). The model could be used as a preliminary filter for fake news detection.

Key words: deception detection, Rhetorical Structure Theory, automated decepted detection, news verification, discourse analysis

1. Introduction

In the contemporary world we deal with the large amount of information that we get from different and diverse sources: newspapers, institutional and non-institutional online media, blogs and social media, TV channels and their websites etc. It is very important to understand the difference between information types and to evaluate reliability of sources. In news reports, rumours, deceptive information and deceptive (fake) news can be easily used for manipulation of public opinion, for information warfare. This is why new tools for automated deception detection and information verification, created for different languages, based on Natural Language Processing methods and models, are required in our society. Now there are no research papers about automated deception detection for the Russian language. There is also a significant lack of linguistics tools for Natural Language Processing which could be helpful in solving the problem. It seems to be a theoretical and methodological challenge.

2. Literature Review

Written texts are a subject of research for studying deception detection methods, especially for English. Digital texts, online reviews (Ott et al., 2011; Mukherjee et al., 2013), fake social network profiles (Kumar, Reddy, 2012), fake dating profiles (Toma, Hancock, 2012) etc. were already investigated. The objective of revealing news verification mechanisms arose rather recently. Fake news may be identified on different levels. Usually researchers tend to combine different levels, from lexics and semantics to syntax. Most studies focus on lexics and semantics and some syntax principles; discourse and pragmatics have still rarely been considered (Rubin et al., 2015) due to the complexity of such approach.

On the lexics level researchers can extract some stylistic features (part of speech, length of words, subjectivity terms etc.) that help to apart tabloid news (they are similar to fake news) with 0.77 accuracy score (Lex et al., 2010). Numbers, imperatives, names of media persons can be extracted from news headlines (Clark, 2014); the numbers of these keywords can be used as features for classification with Support Vector Machines or Naive Bayes Classifier (Lary et al., 2010). Some linguistics markers can be found in lexics and semantics level from the Statement Validity Analysis, the accuracy rate reaches 0.74 (Porter and Juille, 1996). Existing psycholinguistics lexicons, for instance LIWC (Pennebaker and Francis, 1999), can be used in performing binary text classifications for truthful vs deceptive texts (0.70 accuracy rate) (Mihalcea and Strapparava, 1999)—for example, methods can be based on frequency of affective words or action words from lexicons. As to syntax level, Probability Context Free Grammars can be used. Text fragments are presented as a set of rewrite rules to describe syntax structure and produce a parse tree. So we can distinguish rule categories for deception detection with 0.85–0.91 accuracy (Feng et al., 2012). Syntax analysis is often combined with other linguistics or network approaches (Rubin et al., 2015a). On pragmatics level, it is found out that pronouns with antecedents in text are more often used in fake news‘ headlines to pay reader's attention (Blom and Hansen, 2015).

Some studies are focused on creating models that reveal if the described event accords with the facts or not. In (Sauri and Pustejovsky, 2012) authors represent a model, which is based on grammatical fact description structures in English and kindred languages. It has been implemented in De Facto, a factuality profiler for eventualities mentioned in text based on lexical types and syntax constructions. The researchers also created FactBank—annotated corpus in English.

There are three types of fake news: serious fabrications, large-scale hoaxes and humorous fakes (Rubin et al., 2015b). We should also take into consideration the recent research (Hardalov et al., 2016): it proposes the approach for automatically distinguishing credible from fake news, based on different features: linguistic (n-gram), credibility-related (capitalization, punctuation, pronoun use, sentiment polarity), and semantic (embeddings and DBPedia data) features. The accuracy is from 0.75 to 0.99 on three different datasets, but, although the approach, based on combining different levels, is promising, it is hard to compare the results because they are got mainly on news stories which may be considered as hoaxes and fakes (7038 fake texts), not as intentiallly fabricated "serious" news (68 fake texts).

Recent research projects are dedicated to discourse differences between deceptive (fabricated) and truthful (authentic) news, specifically in terms of their rhetorical structures and coherence relation patterns (Rubin et al., 2015). Vector space modeling application lets predict whether a particular news report is truthful or deceptive

(0.63 accuracy) for English. Seriously fabricated news stories were chosen for the dataset. So rhetorical structures and discourse constituent parts and their coherence relations are already reviewed as possible deception detection markers in English news. If we review deception detection methods for other languages, in our case for Russian, we also should keep in mind linguistics and cultural considerations.

RST (Rhetorical Structure Theory) framework (Mann and Thompson, 1988) is addressed to the discourse level of text. It represents text as an hierarchical tree. Some parts are more essential (nucleus) than others (satellite). Elementary discourse units are connected to each other according to relations: elaboration, justify, contrast etc. The theory pretends to be universal for all languages, therefore we chose it for our research. It is used in Russian computational linguistics. Nevertheless automated parser was never worked out specially for the Russian language, it causes constraints in using RST framework in applications.

Support Vector Machines (SVM) method can be grasped as supervised learning models for classification tasks in machine learning. In our case, news reports are shown as vectors in n-dimensional space. A news report is placed in one of two groups, deceptive or truthful. Random forest is as a learning method operates by constructing a multitude of decision trees at training time and, in case of classification task, outputting the class.

3. Research Objective

Our hypothesis is that there are significant differences between structures of truthful news reports and deceptive ones. Our aim is to reveal them using RST relations as deception detection markers, based on the definite corpus. Firstly, we would like to find out what the features from the Rhetorical Structure Theopy should look like: we should detect if RST relation types‘ frequencies, relations’ sequences are important. Then we shall estimate the impact of these features into the successful detection. We shall classify the texts, based on the RST relations labeling, and we shall do our best to predict if news reports are truthful or deceptive.

This model can be useful for news verification, in detecting and filtering deceptive (fake) news. Especially it is of vital necessity for the Russian language, because news reports in Russian nowadays often contain deceptive information and deliberate misinformation, and there is no way how to check it excepting the manual one. Our research is based on the methodology of the news reports research for the English language (Rubin et al., 2015), but it also takes into consideration some features of this research field for Russian.

4. Data Collection

The main difficulty of collecting data set for deception detection is the lack of sources in Russian that contain verified samples of fake and truthful news. There are no Factbanks in Russian, there are no objective, impersonal fact checking websites that contain the reports of investigative journalism. Therefore, the only way out in solving the problem was the reliance on the presented facts, on the factuality. The daily manual monitoring of news lasted 11 months (June 2015-April 2016). Online newspapers in Russian were used as sources. In order to maintain balance we took texts from different sources: well-known news agencies' websites, local or topic-based news portals, online newspapers from different countries (Russia, Ukraine, Armenia etc.). News source mention was not included in corpus text annotations to avoid subjectivity. Blog texts, social media content, news reports based on opinions (not on facts) were excluded from the monitoring. So we used news reports about facts and not analytic journalism stories where different viewpoints are conventional. News stories were analyzed in retrospect when the factuality was already known. Fake reports were put to negative class ('0'), truthful reports were put to positive class ('1').

For instance, news story about airplane crush which appeared only in one source and did not fit facts was considered as fake. News story about death of famous person was condidered as fake after refutation. Airplane accident which was mentioned in diverse sources and confirmed with facts was considered as true. Death of famous person which was confirmed in other news sources by this person‘s friends and relatives was considered as true. So we see two news ’“pairs" about definite topics. They can be not only about the same topic, but about the same event: for example, news story about the Shengen visa centres closing for Russian citizens was considered as fake because at the same time we could see the truthful news story about new rules of document executions and possible delays.

As to news reports with mutual contradictions, a report was added to fake cases if we could see the opposite news reports at the same time in different online media: with some unproven facts and with their refutation which was truthful. It means that if we saw a fake news story we considered the time when it appeared: if there were stories with refutation at the same time, we considered that it was an intended fake and not a journalist's mistake caused by lack of facts.

There are three types of fake news: serious fabrications, large-scale hoaxes, humorous fakes (Rubin et al., 2015b). We analyzed only the first two types, because we are interested in deceptive news that look similar to truthful news. We suggest: if only a report is intended as a fake one, its rhetorical structure differs from a truthful one. That's why we did not add reports, based on author's inaccuracy and not on author's intention, to our corpus.

Generally, the final data set consists of news reports dedicated to 38 different topics, with equal number of truthful and deceptive news stories to each topic, and not more than 12 news reports about the same topic. Each topic was analyzed carefully to define a fake part in the case and to avoid subjectivity and biased evaluation.

5. Corpus Details

The corpus contains 134 news reports, with average length 2700 symbols. Average number of rhetorical relations in text is 17.43. The whole number of rhetorical relations in corpus is 2340. Clauses were taken as elementary discourse units.

For comparison, the dataset in the paper describing the research on which we base our research (Rubin et al., 2015) includes 144 news reports. Corpus in the research about the impact of discourse markers in argument units classification (Eckle-Kohler et al., 2015) consists of 88 documents, predominantly news texts. So the corpus size is conventional for our goals for the initial research on the field of discourse analysis.

There are no discourse parsers for Russian, that‘s why tagging and validation were made manually. We used UAM CorpusTool for discourse-level annotation. We based the research on the ’‘classic’‘ set by Mann and Thompson and added to it the relational categories from extended sets. News reports usually have a definite template, thus, we used a relatively small number of different relational categories. We created relation types Evidence 1 (the source of information, the speaker, is mentioned precisely without hyperlink), Evidence 2 (the source is mentioned imprecisely: «Some experts/media say that…»), Evidence 3 (the source is mentioned precisely with hyperlink) and Evidence 4, the most rare one (the source is mentioned with hyperlink, but the information in the source text does not correspond to the information in the news report). They have the same structure in text, but we guessed that there could be a difference between truthful and deceptive news. Finally we had 33 relation types: ’Circumstance‘, ’Reason‘, ’Evidence1‘, ’Evidence2‘, ’Evidence3‘, ’Evidence4‘, ’Contrast‘, ’Restatement‘, ’Disjunction‘, ’Unconditional‘, ’Sequence‘, ’Motivation‘, ’Summary‘, ’Comparison‘, ’Non-Volitional Cause‘, ’Antithesis‘, ’Volitional Cause‘, ’Non-Volitional Result‘, ’Joint‘, ’Elaboration‘, ’Background‘, ’Solution‘, ’Evaluation‘, ’Interpretation‘, ’Concession‘, ’Means‘, ’Conjunction‘, ’Volitional Result‘, ’Justify‘, ’Condition‘, ’Exemplify‘, ’Otherwise‘, ’Purpose'.

6. Inter-annotator Consistency

We faced the following discrepancies during our tagging work: Background/Sequence/Elaboration; Reason/Unvolitional Cause/Volitional Cause; Purpose/ Unvolitional Result/Volitional Result; Evaluation/Interpretation; Antithesis/Contrast; Elaboration/Justify/Restatement in quotations. We prepared guidelines in our tagging manual for these cases.

The assignment of RST relations is often criticized because it could be connected with the subjectivity of annotators' interpretation: the same text could be annotated in different ways. We tried to solve this problem by preparing a precise manual for tagging and by developing consensus-building procedures. News topics for coders were selected randomly, after that coder A analyzed 66 reports, coder B analyzed the remaining 68 reports. Truthful and deceptive news reports about the same event were annotated by the same person. Therefore, if there could be a variance in segmenting a text into clauses or in tagging a definite rhetorical relation type, similar parts and mutual quotations in truthful and deceptive texts would be annotated in the same way.

We selected Krippendorff’s unitized alpha as a measure to apply because it suits if coders have different approaches to segmenting and labeling in definite text sequences. After the second step the agreement reached 0.75.

7. Data Analysis

The first experiment allows to define a baseline on the lexics level: we decided to choose frequency of lemmas from a sentiment lexicon as a feature for each text. We suggested that it could help identify truthful and deceptive news reports because positive and negative opinion words could be considered as affective words and could replace causation in deceptive texts. We used a list of 5000 sentiment words got from reviews devoted to various topics (Chetviorkin and Loukachevitch, 2012).

The second experiment was run on three different datasets. RST relation types frequencies and their collocations are represented as features. The first dataset (model A) is based on a statistics file which contains data about types of RST relations and their frequencies for each news report. In fact, we deal here with a 'bag of relation types', disregarding their order. As rhetorical structure is tree-like and not flat, we added count of bigrams and trigrams of RST types (based on class nltk.util.ngrams in NLTK 3.0 (Natural Language Toolkit) for Python) for each text in model A to create model B. Model C also contains model A, but in this case it is combined for each news report with count of occurences of top 20 bigrams of RST types and top 20 trigrams of RST types from the whole corpus (here we used module nltk.collocations, threshold not less than 3 occurencies for the whole corpus).

We selected two supervised learning methods for texts classification and machine learning: Support vector machines (SVMs) and Random Forest, both realized in scikit-learn library for Python. SVMs were used with linear kernel and with rbf kernel. In both experiments we used 10-fold cross-validation for estimator performance evaluation.

We also held an additional experiment: the corpus was annotated manually to compare machine learning results, which are based on RST-features, with human asessments. 25 participants, aged 20–35, who did not participate in choosing texts for the coprus or annotating RST relations, marked per e-mail each news report as truthful/fake one (every participant marked all texts). We did not use online forms, because these people also gave expert interviews during preliminary qualitative sociological research about fake news perception, and it was convenient to discuss all issues per e-mail. After that we counted common scores.

8. Statistical Procedures

The results for the first and second experiments are presented in Table 1. We can evaluate that the classification task is solved better by SVMs (linear kernel) for model A, without addition of bigrams and trigrams features. The accuracy score is 0.65. It means that the sequence of RST relations is not so important as the frequencies of RST relation types. The score can be compared with the predictive power of the model for English (Rubin et al., 2015) which is 0.63. It is also more than the human ability score to detect deceptive information (0.54) which was got in different experiments listed in the article (Rubin et al., 2015). The results for our additional experiment with manual tags are got together in Table 2. They can be compared with the results for English. They show less recall and less precision than the results of automated deception detection for Russian in our case.

The most significant features which influence on linear SVMs classification for model A are: 'Justify', 'Evidence3', 'Contrast', 'Evidence1', 'Volitional Cause', 'Comparison'. So we decided correctly to divide 'Evidence' into 4 types. Student's t-test to check the statistical significance of these six features showed that first five ones are significant, about 'Comparison' we cannot state the same with confidence (p-value measure 0.07858).

'Volitional Cause' is one of the most significant features, and this relation type is more typical for deceptive texts. Probably this could be explained so: authors of fake news pay more attention to the causation, because they want to explain an event with the internal logic of their position, without any inconsistencies. 'Circumstance' and 'Elaboration' are also more typical for deceptive news reports, and they also point to the logical structure of a text. Herewith, 'Volitional Cause' is not the most significant feature. 'Justify', 'Evidence3' and 'Evidence1', 'Contrast' are more typical for thuthful texts. Hence, truthful news reports contain more often information with rational, precise source mention and direct link to it (whereas 'Evidence2' is more typical for fake news, as it contains imprecise source mention). The presence of 'Contrast' and 'Comparison' among important features can be explained so: truthful news reports in our corpus can be considered as rebuttals of fake news reports, therefore they refer to them and contain parts of deceptive texts. 'Contrast' and 'Comparison' could be used as a link between a deceptive text citation and an explanation why it is a fake.

Table 1. Results for different classifiers

	Precision	**Accuracy**	**Recall**	**F-measure**
Support Vector Machines, rbf kernel, 10-fold cross-validation				
Baseline	0.38	0.42	0.54	0.42
Model A	0.54	0.53	0.51	0.51
Model B	0.60	0.55	0.52	0.50
Model C	0.65	0.61	0.56	0.57
Support Vector Machines, linear kernel, 10-fold cross-validation				
Baseline	0.23	0.37	0.49	0.31
Model A	0.64	0.65	0.65	0.63
Model B	0.64	0.60	0.48	0.53
Model C	0.62	0.59	0.60	0.59
Random Forest Classifier, 10-fold cross-validation				
Baseline	0.48	0.48	0.55	0.49
Model A	0.56	0.54	0.45	0.47
Model B	0.60	0.63	0.56	0.56
Model C	0.57	0.55	0.46	0.49

Table 2. Manual (human) asessments for news reports

	Precision	**Recall**	**F-measure**
Scores for human asessments	0.55	0.46	0.50

9. Discussion

Automated deception detection based on the Rhetorical Structure Theory seems to be a promising and methodologically challenging research topic, and further measures should be taken to find features for deception/truth detection in automated news verification model for the Russian language. Our hypothesis is confirmed. The present research is initial, and the model should be developed and modified, learned and tested on larger data collections with different topics. In addition, we should use a complex approach and combine this method with other linguistics and statistical methods. For instance, syntactic level features on top of sequences of discourse relations should be studied. Discourse markers may be also taken into consideration as separate features. The guidelines for gathering a training corpus of obviously truthful/deceptive news should also be improved.

The extrapolation of the existing model to all possible news reports in Russian, devoted to different topics, would be incorrect. But despite this fact, it can already be used as a preliminary filter for deceptive (fake) news detection. Results of its work should be double-checked and refined, especially for suspicious instances fact checking.

We tried to take into consideration 'the trees'—hierarchies of RST relation types in texts and dependences between relation types. This aspect should be studied more deeply and intensively.

The model is also restricted by the absence of automated discourse parser for Russian. It is typical for other Natural Language Processing tasks for Russian which deal with RST.

Finally, the assignment of RST relations to news report could be connected with the subjectivity of annotators' interpretation. Despite of inter-annotator consistency measures, this problem exists and could be partly solved by preparing more precise manuals for tagging and by developing consensus-building procedures.

10. Conclusions

News verification tends to be a very important issue in our actual world, with its information warfare and propaganda methods. The precision of human deception detection ability for news reports in the present research in Russian is 0.55.

We collected a corpus (134 news reports, truthful and fake ones). We segmented the texts manually and applied RST relations tagging to them. As to the experiments, three dataset models for machine learning were based on features from the Rhetorical Structure Theory. We also used the model based on features from the sentiment lexicon as a baseline. We applied Support vector machines (SVMs) algorithm (linear kernel / rbf kernel) and Random Forest to classify the news reports into 2 classes: truthful/deceptive. The predictive power of the model based simply on frequencies of RST relation types in texts is the highest one (the sequence of RST relations is not so important). The classification task is solved better by SVMs (linear kernel) for this dataset (0.65 accuracy score). Such RST relation types as Justify, Evidence3, Contrast, Evidence1, Volitional Cause, Comparison produce the most significant features. The modified model could combine RST relations markers with other deception detection markers in order to make a better predictive model.

References

1. *Blom J. N., Hansen K. R.* (2015), Click bait: Forward-reference as lure in online news headlines, Journal of Pragmatics 76, pp. 87–100.
2. *Chetviorkin I. I., Loukachevitch N. V.* (2012), Extraction of Russian Sentiment Lexicon for Product Meta-Domain, Proceedings of COLING 2012: Technical Papers, pp. 593–610.
3. *Clark R.* (2014), Top 8 Secrets of How to Write an Upworthy Headline, Poynter, URL: http://www.poynter.org/news/media-innovation/255886/top-8-secrets-of-how-to-write-an-upworthy-headline/
4. *Eckle-Kohler J., Kluge R., Gurevych I.* (2015), On the Role of Discourse Markers for Discriminating Claims and Premises in Argumentative Discourse, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2236–2242.
5. *Feng S., Banerjee R., Choi Y.* (2012), Syntactic Stylometry for Deception Detection, Proceedings 50th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Volume 2: Short Papers, pp. 171–175.
6. *Hardalov M., Koychev I., Nakov P.* (2016), In Search of Credible News, Artificial Intelligence: Methodology, Systems, and Applications, pp. 172–180.
7. *Kumar N., Reddy R. N.* (2012), Automatic Detection of Fake Profiles in Online Social Networks, BTech Thesis.
8. *Lary D. J., Nikitkov A., Stone D.* (2010), Which Machine-Learning Models Best Predict Online Auction Seller Deception Risk?, American Accounting Association AAA Strategic and Emerging Technologies.
9. *Lex E., Juffinger A., Granitzer M.* (2010), Objectivity classification in online media, Proceedings of the 21st ACM conference on Hypertext and hypermedia, pp. 293–294.
10. *Mann W. C., Thompson S. A.* (1088), Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, Text, vol. 8, no.3, pp. 243–281.
11. *Mihalcea R., Strapparava C.* (1999), The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language, Proceedings 47th Annual Meeting of the Association for Computational Linguistics, Singapore, pp. 309–312.
12. *Mukherjee A. et al.* (2013), Fake Review Detection: Classification & Analysis of Real & Pseudo Reviews. Technical Report, Department of Computer Science, University of Illinois at Chicago, & Google Inc.
13. *Ott M., Choi Y., Cardie C., Hancock J. T.* (2011), Finding deceptive opinion spam by any stretch of the imagination. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 309–319.
14. *Pennebaker J., Francis M.* (1999), Linguistic inquiry and word count: LIWC, Erlbaum Publishers.
15. *Porter S., Juille J. C.* (1996), The language of deceit: An investigation of the verbal clues to deception in the interrogation context, Law and Human Behavior, vol. 20, № 4, pp. 443–458.
16. *Rubin V. L., Conroy N. J., Chen Y. C.* (2015), Towards News Verification: Deception Detection Methods for News Discourse, Proceedings of the Hawaii International Conference on System Sciences (HICSS48) Symposium on Rapid

Screening Technologies, Deception Detection and Credibility Assessment Symposium, January 5–8, Grand Hyatt, Kauai, 11 pages.

17. *Rubin V. L., Conroy N. J., Chen Y.* (2015a), Automatic Deception Detection: Methods for Finding Fake News, Conference: ASIS T2015, At St. Louis, MO, USA.
18. *Rubin V. L., Conroy N. J., Chen Y.* (2015b), Deception Detection for News: Three Types of Fakes. Conference: ASIS T2015, At St. Louis, MO, USA.
19. *Sauri R., Pustejovsky J.* (2012), Are You Sure That This Happened? Assessing the Factuality Degree of Events in Text, Computational Linguistics, pp. 1–39.
20. *Toma C. L., Hancock J. T.* (2012), What Lies Beneath: The Linguistic Traces of Deception in Online Dating Profiles, Journal of Communication, vol. 62, № 1, pp. 78–97.

TOWARDS BUILDING A DISCOURSE-ANNOTATED CORPUS OF RUSSIAN

Pisarevskaya D. (dinabpr@gmail.com)[1],
Ananyeva M. (ananyeva@isa.ru)[2],
Kobozeva M. (kobozeva@isa.ru)[2],
Nasedkin A. (kloudsnuff@gmail.com)[3],
Nikiforova S. (son.nik@mail.ru)[3],
Pavlova I. (ispavlovais@gmail.com)[3],
Shelepov A. (alexshelepov1992@gmail.com)[3]

[1]Institute for System Programming of the RAS, Moscow, Russia;
[2]Institute for Systems Analysis FRC CSC RAS, Moscow, Russia;
[3]NRU Higher School of Economics, Moscow, Russia

For many natural language processing tasks (machine translation evaluation, anaphora resolution, information retrieval, etc.) a corpus of texts annotated for discourse structure is essential. As for now, there are no such corpora of written Russian, which stands in the way of developing a range of applications. This paper presents the first steps of constructing a Rhetorical Structure Corpus of the Russian language. Main annotation principles are discussed, as well as the problems that arise and the ways to solve them. Since annotation consistency is often an issue when texts are manually annotated for something as subjective as discourse structure, we specifically focus on the subject of inter-annotator agreement measurement. We also propose a new set of rhetorical relations (modified from the classic Mann & Thompson set), which is more suitable for Russian. We aim to use the corpus for experiments on discourse parsing and believe that the corpus will be of great help to other researchers. The corpus will be made available for public use.

Keywords: rhetorical structure theory, discourse analysis, corpus linguistics, corpus annotation, discourse structure, inter-annotator agreement

ПРИНЦИПЫ РАЗРАБОТКИ ДИСКУРСИВНОГО КОРПУСА РУССКОГО ЯЗЫКА

Писаревская Д. (dinabpr@gmail.com)[1],
Ананьева М. (ananyeva@isa.ru)[2],
Кобозева М. (kobozeva@isa.ru)[2],
Наседкин А. (kloudsnuff@gmail.com)[3],
Никифорова С. (son.nik@mail.ru)[3],
Павлова И. (ispavlovais@gmail.com)[3],
Шелепов А. (alexshelepov1992@gmail.com)[3]

[1]Институт системного программирования РАН, Москва, Россия; [2]Институт системного анализа ФИЦ ИУ РАН, Москва, Россия; [3]НИУ ВШЭ, Москва, Россия

1. Introduction

Discourse analysis is the linguistics level that deals with language units of maximal size (Kibrik, Podlesskaya, 2009, 26). During discourse analysis the text is often represented as a hierarchical tree with its parts connected by various rhetorical relations. Discourse and pragmatics have been considered in Natural Language Processing only in recent years due to the complexity of the approach. Discourse parsing can be used in a wide range of natural language processing tasks, including machine translation evaluation, sentiment analysis, information retrieval, text summarization, information extraction, anaphora resolution, question-answering systems, text classification, etc.—it gives significant performance gain in all these applications, as has been shown by a lot of research.

Creation of corpora with discourse structure has become very popular in recent years because they are then used for developing machine learning algorithms to build automated systems for discourse parsing and analysis. Discourse parsers already exist for several languages, most notably for English (RASTA, SPADE, HILDA, CODRA parsers). However, there are no discourse-annotated corpora for written Russian at the moment, and therefore no possibility of creating an automated discourse parser, and as long as only manual annotation of texts is possible, discourse analysis will not be used in any applications for Russian. That is why it is essential to develop a publicly available discourse-annotated corpus for the Russian language.

In this paper we describe the first steps of building a discourse corpus of Russian: the annotation procedure, including establishing the appropriate set of discourse relations, the process of measuring the inter-annotator agreement, and various challenges we faced along the way.

1.1. Background

There are different approaches to discourse analysis. In Rhetorical Structure Theory (RST) discourse structure amounts to a non-projective tree. Penn Discourse Treebank (PDTB) style is connective-led (PDTB (Webber et al., 2016), TurkishDB (Zeyrek et al., 2013), etc.) or punctuation-led (Chinese Discourse TreeBank (Zhou, Xue, 2015)) and is not presented in a tree form. Models based on cohesive relations (Halliday, Hasan 1976) are also not tree-like. We decided to choose RST to take into consideration not only cohesive markers and discourse cues, but also discourse structure of texts. It is important, for example, for coreference resolution in English—sometimes the most crucial for it is the rhetorical distance and not the linear one, cf. (Loukachevitch et al. 2011).

Therefore, for our corpus we adopt the RST framework (Mann, Thompson, 1988). It represents text as a hierarchy of elementary discourse units (EDUs) and describes relations between them and between bigger parts of text. Some EDUs are more essential and carry more important information (nucleus) than others (satellite). There are two rhetorical relation types: nucleus-satellite and multinuclear. While the first type connects a nucleus and a satellite, the latter includes EDUs that are equally important in the analysed discourse. The set of rhetorical relations can vary; it can include, for instance, such relations as Elaboration, Justify, Contrast, Antithesis, Volitional Result, etc. The rhetorical structure theory claims to be applicable to all languages.

In our work we take into account the existing experience of constructing discourse corpora. There are many RST-annotated corpora of different languages. The most well-known one is the RST Discourse Treebank (Carlson et al., 2003)—an English-language corpus of Wall Street Journal articles (385 articles—176,383 tokens). It is the biggest discourse corpus with a detailed manual. Potsdam Commentary Corpus (Stede, Neumann, 2014) [http://corpus.iingen.unam.mx/rst/manual_en.html] is a German-language corpus that consists of newspaper materials (175 articles—32,000 tokens). CorpusTCC (Pardo et al., 2004) is a corpus of Brazilian Portuguese. It includes 100 introductions (53,000 tokens) to PhD theses. Well-developed are also corpora for other languages: Dutch—Dutch RUG Corpus (van der Vliet et al., 2011), Basque—RST Basque Treebank (Iruskieta et al., 2013), Chinese and Spanish—Chinese/Spanish Treebank as a parallel corpus (Cao et al., 2016), etc. Different sets of rhetorical relations have been created based on the “classic set” (Mann, Thomspon, 1988). For instance, the RST Discourse Treebank makes use of 88 relation types (53 nucleus-satellite and 25 multinuclear relations), the Potsdam Commentary Corpus is based on 31 relation types.

The only existing discourse corpus project for Russian is TED-Multilingual Discourse Treebank. This project contains a parallel corpus of TED talks transcripts for 6 languages, including Russian (along with English, Turkish, European Portuguese, Polish, and German). However, it is based on the principles of the Penn Discourse Treebank annotations framework—discourse connectives as discourse-level predicates with a binary argument structure at a local level (Prasad et al., 2007; Zeyrek et al., 2013)—and not on the RST framework. Besides, this recent effort is still in progress and is not publicly available yet.

The foundation for the project of the Discourse-annotated corpus of Russian was laid by the following works of the research team of the Institute for Systems Analysis, FRC CSC RAS (Ananyeva, Kobozeva, 2016 [1, 2]).

2. Rhetorical Structure Corpus for the Russian Language

The Discourse-annotated corpus of Russian will include texts of different genres (science, popular science, news stories, and analytic journalism). The development of the corpus will be continued for 3 years, during which time we are going to annotate more than 100,000 tokens. The corpus will be available for public use. The user will be able to view annotated texts (represented as discourse trees), search for specific relations (or sequences thereof) and word forms, download the annotated texts in XML format.

2.1. Annotation Principles

After conducting extensive research on discourse corpora of other languages, we have developed a detailed annotation manual. As a tool for annotation we have chosen an open-source tool called rstWeb [https://corpling.uis.georgetown.edu/rstweb/info/], which allows to edit a set of relations and change other features if needed.

International experience of discourse annotation demonstrates that due to grammatical differences between languages, an adaptation of the classic RS theory is necessary for almost all of them. That is why in our project we will, among other things, aim to specify the concept of a discourse unit and the set of rhetorical relations for Russian.

Firstly, we have established a preliminary notion of an elementary discourse unit, which, from a syntactic point of view, we take to be roughly equivalent to a clause (similarly to the classic Mann & Thompson approach). However, there are several notable exceptions, such as nominalization constructions with prepositions like *для* ‘for’ and *из-за* ‘because of’ being classified as an EDU and relative clauses with restrictive semantics not being classified as one.

Secondly, we have discussed main annotation principles and created a detailed manual to guide the annotators. It included description of the following 22 relations, which were based on the “classic set” with the specific features of news and scientific texts in Russian taken into account.

- 16 nucleus-satellite (mononuclear) relations: Background, Cause (with subtypes: Volitional Cause and Non-volitional Cause), Evidence, Effect (with subtypes: Volitional Effect and Non-volitional Effect), Condition, Purpose, Concession, Preparation, Conclusion, Elaboration, Antithesis, Solutionhood, Motivation, Evaluation, Attribution (with subtypes: Attribution1 (precise source specification) and Attribution2 (imprecise source specification)), Interpretation.
- 6 multinuclear relations: Contrast, Restatement, Sequence, Joint, Comparison, Same-unit.

We decided to add Preparation and Conclusion to the set due to the genre properties of scientific and analytic texts. We divided Attribution into two subtypes due to the differing level of precision of specifying the information source in news stories.

There are two strategies of annotators’ work in RST analysis (Carlson et al., 2003). An annotator could apply relations to the segments sequentially, from one segment to another, connecting the current node to the previous node (left-to-right). This method is suitable for short texts, such as news reports, but even in such texts there is a risk of overlooking important relations. The other method is more flexible: the annotator segments multiple units simultaneously, then builds discourse sub-trees for each segment, links nearby segments and builds firstly larger subtrees and after that the final tree, linking key parts of the discourse structure (top-down and bottom-up). It is more suitable for big texts. We chose the second method of tagging since it is more intuitive and easier for the annotator.

For the first 3 texts annotators used the set of discourse relations specified above. The texts were of approximately the same length (34, 26 and 38 sentences respectively). All of them were short news articles. The annotators followed the initial guidelines while annotating pilot texts: they segmented the texts and assigned RST relations to the resulting segments. During subsequent discussion it has become clear that this set of relations was not quite convenient for the annotation since some of the relations were extremely hard to differentiate between. Moreover, we have realized that adopting a "classic set" requires further modifications as some relations are probably more obvious and therefore more common in English than in the Russian language.

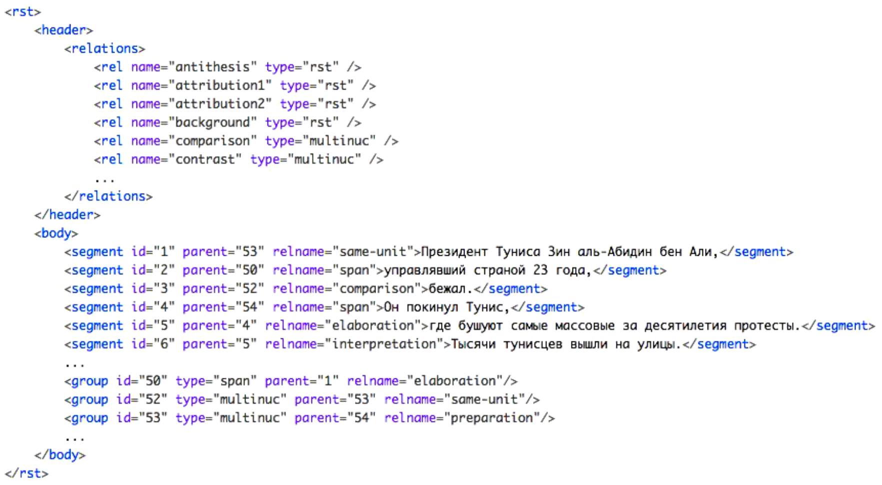
2.2. Inter-annotator agreement measurement

One of the main problems with RST tagging is the subjectivity of annotators' interpretation: the same text can be annotated in very different ways (Artstein, Poesio, 2008). However, a simple discussion is not enough to establish the level of inter-annotator agreement (IAA). It must be measured quantitatively using a valid and reliable statistic.

Much like in other discourse analysis tasks (Miltsakaki et al., 2004; Eckle-Kohler, 2015), we faced certain challenges with inter-annotator consistency computation. Although the rules of splitting text into EDUs are relatively straightforward, the resulting segmentation is rarely the same for any text segmented by different people. That is why, for example, the Cohen's kappa coefficient is not suitable in our case. The token-based Fleiss' kappa is also not applicable as we deal with units that consist of several tokens. We have finally selected Krippendorff's unitized alpha as a statistic to measure inter-annotator agreement. It operates on whole annotation spans instead of isolated tokens, it can be calculated for any number of annotators, it can be applied to sparse data, and it can process features of different types, including nominal features in our case. As splitting text into EDUs and labeling relations are two separate tasks, the inter-annotator agreement can be measured separately as well. However, Krippendorff's unitized alpha can (and will in our case) be used for both measurements.

The corpus size used for inter-annotator consistency calculation varies from one project to another. Usually it covers about 30 units (Lacy, Riffe, 1996), but we decided to take texts that contain more units so we could check if relation types in the manual are suitable for further work. The total number of EDUs was approximately 190.

The RST tagging by means of rstWeb tool, which is used by annotators, is done in the browser (see Fig. 2), but the system allows to export the result file as an xml-document, which has the following structure:



```
<rst>
    <header>
        <relations>
            <rel name="antithesis" type="rst" />
            <rel name="attribution1" type="rst" />
            <rel name="attribution2" type="rst" />
            <rel name="background" type="rst" />
            <rel name="comparison" type="multinuc" />
            <rel name="contrast" type="multinuc" />
            ...
        </relations>
    </header>
    <body>
        <segment id="1" parent="53" relname="same-unit">Президент Туниса Зин аль-Абидин бен Али,</segment>
        <segment id="2" parent="50" relname="span">управлявший страной 23 года,</segment>
        <segment id="3" parent="52" relname="comparison">бежал.</segment>
        <segment id="4" parent="54" relname="span">Он покинул Тунис,</segment>
        <segment id="5" parent="4" relname="elaboration">где бушуют самые массовые за десятилетия протесты.</segment>
        <segment id="6" parent="5" relname="interpretation">Тысячи тунисцев вышли на улицы.</segment>
        ...
        <group id="50" type="span" parent="1" relname="elaboration"/>
        <group id="52" type="multinuc" parent="53" relname="same-unit"/>
        <group id="53" type="multinuc" parent="54" relname="preparation"/>
        ...
    </body>
</rst>
```

Fig. 1. XML structure of an annotated file

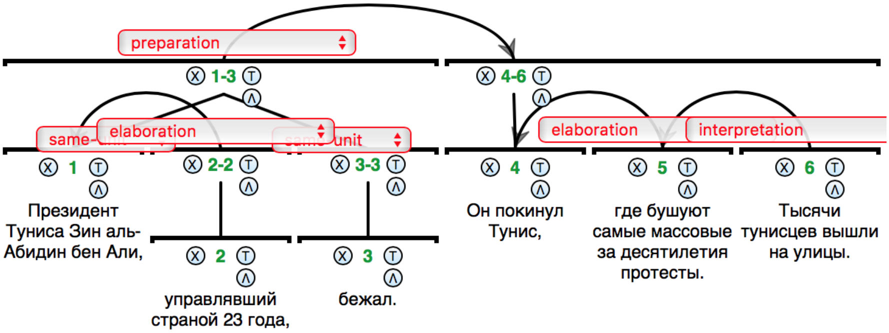

preparation
1-3
4-6
same-unit
elaboration
same-unit
elaboration
interpretation
1
Президент Туниса Зин аль-Абидин бен Али,
2-2
3-3
2
управлявший страной 23 года,
3
бежал.
4
Он покинул Тунис,
5
где бушуют самые массовые за десятилетия протесты.
6
Тысячи тунисцев вышли на улицы.

Fig. 2. Annotation in the browser

All the relations used in the scheme are listed in the header of the xml-document.

Each EDU tag includes two ids and a relation type, where "segment id" stands for the id of the EDU, "parent"—for the id of the nucleus in case it is a nucleus-satellite relation and "relname"—for the type of the relation. If the relation is multinuclear, "segment" and "parent" ids both represent the ids of equal by discourse importance EDUs. If the relation type is specified as "span", the EDU is included in a bigger discourse group which is assigned a new id (i.e. the EDUs 4–6 form a bigger group of relations and the EDU 4 as the main nucleus in this group is marked to have a parent with id 54 which is automatically assigned to this group: <segment id="4" parent="54" relname="span">).

Calculation of the IAA coefficient was implemented in Python. Xmltodict 0.10.2 package was used to read and to convert the XML-object of the marked-up text to the Python dictionary. The code used for IAA calculation can be accessed via GitHub [https://github.com/nasedkinav/rst_corpus_rus/blob/master/krippendorffs_alpha.py].

Since the ids of segments and groups may differ in the texts annotated by different people, we have decided to use concatenated text spans to uniquely identify the selected relations since it is the only reliable data between distinct annotations. This format has the additional advantage because it allows to locate identical relations in different parts of text in case of different EDU fragmentation.

During the first iteration of the particular annotator's markup processing, each of the relations trees is traversed in such a way that each node is associated with an ordered by id set of segments of the text, dominated by the node. The "span" relations were not counted during the IAA measurement since this relation plays a structural role in annotation and has no actual meaning. After that, the index of the form {key: value} was produced for all the relations, where the value is the type of the relation, and the key is represented as a string:

- for mononuclear relations: "nuclear: <nuclear_text>, satellite: <satellite_text>"
- for multinuclear relations: "multinuclear: <multinuclear_text>",

where <nuclear_text>, <satellite_text> and <multinuclear_text> are replaced by correspondent parts of the text. After performing this procedure for each of the annotators, the obtained indices are combined by key <key>, and the list of all the values of the relation, marked by each annotator, is assigned to it. Length of the list can be lower than the number of the annotators when the relation is absent in somebody's markup.

According to (Krippendorff, 2013) we then build the reliability data matrix:

	key_1	key_2	…	key_u	…	key_N
obs_1	$value_{key_1,obs_1}$	$value_{key_2,obs_1}$	…	$value_{key_u,obs_1}$	…	$value_{key_N,obs_1}$
obs_2	$value_{key_1,obs_2}$	$value_{key_2,obs_2}$	…	$value_{key_u,obs_2}$	…	$value_{key_N,obs_2}$
…	…	…	…	…	…	…
obs_m	$value_{key_1,obs_m}$	$value_{key_2,obs_m}$	…	$value_{key_u,obs_m}$	…	$value_{key_N,obs_m}$
Number of coders marked key_u	m_1	m_2	…	m_u	…	m_N

where key_u serves as encoding unit and obs_i stands for particular annotator. Using this matrix, the coincidence matrix within units is calculated (Krippendorff, 2013):

$$\begin{array}{c|cccc|c}
 & 1 & \cdots & k & \cdots & \\
\hline
1 & o_{11} & \cdots & o_{1k} & \cdots & n_1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c & o_{c1} & \cdots & o_{ck} & \cdots & n_c = \sum_k o_{ck} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\hline
 & n_1 & \cdots & n_k & \cdots & n = \sum_c \sum_k o_{ck}
\end{array}$$

where k, c are concrete relation types and

$$o_{ck} = \sum_u \frac{number\ of\ c - k\ pairs\ in\ unit\ key_u}{m_u - 1},$$

where u is the encoded unit (key_u), m_u is the number of annotators who have marked up this unit.

The final calculation of the coefficient can be done in the following way:

$$\alpha_{nominal} = 1 - \frac{D_o}{D_e} = \frac{A_o - A_e}{1 - A_e} = \frac{(n-1)\sum_c o_{cc} - \sum_c n_c\,(n_c - 1)}{n(n-1) - \sum_c n_c\,(n_c - 1)}$$

Fig. 3. Coefficient calculation

We have measured the IAA coefficient for each of three texts and the coefficients for the texts were 0.2792, 0.3173 and 0.4965 respectively. We suppose the third text has the higher IAA coefficient due to the easier and more obvious discourse structure.

The acceptable level of Krippendorff's unitized alpha coefficient for our task would be approximately 0.8 and our results for every text were much lower.

2.3. Initial tree's modification

We have decided to reduce the set of RST relations used for annotation in order to reach the higher IAA coefficient and to minimize the subjectivity of the annotation.

One of the main reasons to exclude particular relations was their high specificity and low frequency of their usage during annotation. Although presence of such relations would not radically affect IAA, reducing the relations' set would make the annotation task easier, and at the same time we would not lose much if we got rid of highly specific and rare relations. If there was always a possibility of replacing some relation with another, more common one, without a great loss in semantic adequacy, it was considered to be an argument in favor of excluding it. The changes that we have accepted after a thorough analysis and much discussion are listed below.

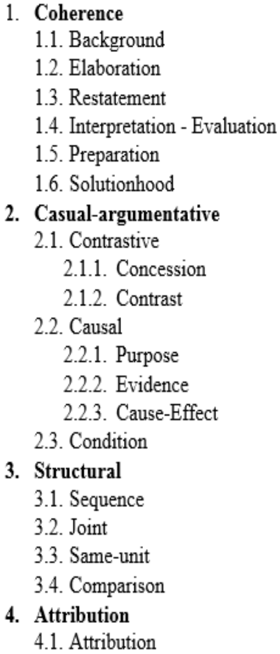
We have decided to exclude from the set of relations

- Motivation, since it is very specific and therefore extremely rare: it was used only 2 times in these three texts (approx. 190 EDUs).
- Antithesis (nucleus-satellite relation), since the only difference between Antithesis and Contrast (multinuclear relation) is that in Antithesis one part should be more important than the other. None of the annotators could establish the relative importance of EDUs in such cases.
- Volitional and Non-Volitional subtypes of Cause and Effect, since in many cases it was impossible to determine whether the actions were intentional or not. However, this distinction might be important for some of the tasks the corpus will be needed for. Those who will use the corpus for this kind of tasks will have the opportunity to substitute Cause/Effect relation with Volitional Cause/Effect or Non-volitional Cause/Effect themselves (as the annotated texts will be available for downloading in an easily changeable XML format).
- Conclusion, because it is quite rare and can be considered a subtype of Restatement, which we decided to use for contexts when the Conclusion relation could be possible.

We have combined in one relation

- Cause and Effect, since the difference between the two lies in determining the nucleus, which is cause in the Cause relation and effect in the Effect relation. Thus, the annotator has to conclude what is more important in two given EDUs: the cause or the effect, which is very subjective.
- Interpretation and Evaluation, since the difference between these relations is very subtle and in order to distinguish between them, one has to determine the degree of objectivity of the evaluation, and that is again very subjective.
- Attribution1 and Attribution2, since the level of precision required for Attribution1 is often unstable and unclear.

All of the above has resulted in a new RST relations tree. The set of relations in Fig. 4 is final and will be used during the rest of the annotation process:



1. **Coherence**
   1.1. Background
   1.2. Elaboration
   1.3. Restatement
   1.4. Interpretation - Evaluation
   1.5. Preparation
   1.6. Solutionhood
2. **Casual-argumentative**
   2.1. Contrastive
      2.1.1. Concession
      2.1.2. Contrast
   2.2. Causal
      2.2.1. Purpose
      2.2.2. Evidence
      2.2.3. Cause-Effect
   2.3. Condition
3. **Structural**
   3.1. Sequence
   3.2. Joint
   3.3. Same-unit
   3.4. Comparison
4. **Attribution**
   4.1. Attribution

Fig. 4. Final set of relations

After modifying the set of discourse relations, three new texts were annotated and the IAA was measured again. The texts were, respectively, 37, 44 and 28 sentences long and all of them were short news articles, same as during the previous IAA measurement. The new IAA coefficients were 0.7768, 0.691 and 0.7615 respectively, which indicates a big leap in the annotation quality. These three texts, annotated in XML format, are available at [https://github.com/nasedkinav/rst_corpus_rus] along with other texts annotated so far. The web interface for the corpus will be created as soon as the appropriate number of texts (and tokens) is reached.

3. Conclusion

By establishing a reliable set of discourse relations we have formed a sound basis for further work. The two iterations of IAA measurement let us believe that using the final relation list will lead to a less biased annotation from now on, which is very important because of the well-known subjectivity of the discourse relations' understanding.

During the rest of the project every text will be annotated by one person and then checked but not annotated by another one. We plan to measure IAA regularly to ensure that the agreement level remains high enough.

After annotating approximately one hundred texts, we plan to conduct several experiments regarding automatic EDUs and discourse relations recognition. Automatic rhetorical structure analysis often relies heavily on determining linguistic discourse markers—connectors that join clauses and sentences into an interconnected piece of text. That is why during the annotation we will also fixate and analyze these markers in order to identify particular words and constructions that indicate discourse relations.

Acknowledgements

We would like to express our sincere gratitude to the advisor of the Discourse-annotated corpus of Russian project, Svetlana Toldova, for the continuous support for the project and related research.

References

1. *Ananyeva M. I., Kobozeva M. B.* (2016), Developing the corpus of Russian texts with markup based on the Rhetorical Structure Theory [Razrabotka korpusa tekstov na russkom yazyke s razmetkoj na osnove teorii ritorecheskikh struktur], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2016" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2016"], Moskva, available at: http://www.dialog-21.ru/media/3460/ananyeva.pdf
2. *Ananyeva M. I., Kobozeva M. B.* (2016), Discourse analysis for natural language processing tasks [Diskursivnyi analiz v zadachakh obrabotki yestestvennogo yazyka], Informatics, management and systems analysis: Proceedings of the IV All-Russian Conference for young scientists [Informatika, upravleniye i sistemnyi analiz: Trudy IV Vserossiiskoi nauchnoi konferencii molodykh uchenykh s mezhdunarodnym uchastiyem], Tver', pp. 138–148, available at: http://www.isa.ru/icsa/images/stories/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA_%D0%A2%D0%BE%D0%BC_1.pdf#page=139
3. *Artstein R., Poesio M.* (2008), Inter-coder agreement for computational linguistics. Computational Linguistics 34(4), pp. 555–596.
4. *Cao S. Y., da Cunha I., Iruskieta M.* (2016), Elaboration of a Spanish-Chinese parallel corpus with translation and language learning purposes, 34th International Conference of the Spanish Society for Applied Linguistics (AESLA), to appear.

5. *Carlson L., Marcu D., Okurowski M. E.* (2003), Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory, Current directions in discourse and dialogue, Kluwer Academic Publishers, pp. 85–112.
6. *Eckle-Kohler J., Kluge R., Gurevych I.* (2015), On the Role of Discourse Markers for Discriminating Claims and Premises in Argumentative Discourse, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2236–2242.
7. *Halliday, M. A. K., Hasan R.* (1976), Cohesion in English. London: Longman.
8. *Hayes A. F., Krippendorff K.* (2007), Answering the call for a standard reliability measure for coding data, Communication Methods and Measures Vol. 1, pp. 77–89.
9. *Iruskieta M., Aranzabe M. J., Díaz de Ilarraza A., Gonzalez I., Lersundi M., Lopez de la Calle O.* (2013), The RST Basque TreeBank: an online search interface to check rhetorical relations, IV Workshop RST and Discourse Studies. Fortaleza, Brasil, Outubro 21–23, pp. 40–49.
10. *Kibrik A., Podlesskaya V.* (2009), Stories of dreams: A Corpus-based Research of Russian Oral Discourse [Rasskazy o snovideniyakh: Korpusnoye issledovaniye ustnogo russkogo diskursa], Yazyki slavyanskikh kul'tur, Moskva.
11. *Krippendorff K.* (2013), Computing Krippendorff's Alpha-Reliability, available at: http://web.asc.upenn.edu/usr/krippendorff/mwebreliability5.pdf.
12. *Lacy S., Riffe D.* (1996), Sampling error and selecting intercoder reliability samples for nominal content categories: Sins of omission and commission in mass communication quantitative research, Journalism & Mass Communication Quarterly No 73, pp. 969–973.
13. *Loukachevitch N. V., Dobrov G. B., Kibrik A. A., Khudiakova M. V., Linnik A. S.* (2011), Factors of referential choice: computational modeling [Faktory referencial'nogo vybora: komp'yuternoye modelirovaniye], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2011" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2011"], Moskva, available at: http://www.dialog-21.ru/media/1446/45.pdf
14. *Mann W. C., Thompson S. A.* (1988), Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, Text 8, 3, 1988, pp. 243–281.
15. *Miltsakaki E., Prasad R., Joshi A., Webber B.* (2004), Annotating discourse connectives and their arguments, Proceedings of the HLT/NAACL Workshop on Frontiers in Corpus Annotation, Boston, Massachusetts, USA, pp. 9–16.
16. *Pardo T. A. S., Nunes M. G. V., Rino L. H. M.* (2004), Dizer: An automatic discourse analyzer for brazilian portuguese, Brazilian Symposium on Artificial Intelligence, Springer Berlin Heidelberg, pp. 224–234.
17. *Prasad R., Miltsakaki E., Dinesh N., Lee A., Joshi A., Robaldo L., Webber B.* (2007). The Penn Discourse Treebank 2.0 Annotation Manual. Technical Report 203, Institute for Research in Cognitive Science, University of Pennsylvania.
18. *Stede M., Neumann A.* (2014), Potsdam Commentary Corpus 2.0: Annotation for Discourse Research. Proc. of LREC, Reykjavik.

19. *Van der Vliet N., Berzlanovich I., Bouma G., Egg M., Redeker G.* (2011), Building a Discourse-Annotated Dutch Text Corpus. Proceedings of the Workshop "Beyond Semantics: Corpus-based Investigations of Pragmatic and Discourse Phenomena", Goettingen, Germany, 23–25 February 2011, pp. 157–171.
20. *Webber B., Prasad R., Lee A., Joshi A.* (2016)., A Discourse-Annotated Corpus of Conjoined VPs. Proc. 10th Linguistics Annotation Workshop, Berlin, pp. 22–31.
21. *Zeyrek D., Demirşahin I., Sevdik Çallı A. B., Çakıcı R.* (2013), Turkish Discourse Bank: Porting a discourse annotation style to a morphologically rich language. Dialogue and Discourse, 4(2), pp. 174–184.
22. *Zhou Y., Xue N.* (2015), The Chinese Discourse TreeBank: A Chinese corpus annotated with discourse relations. Language Resources and Evaluation, pp. 397–431.

BRIDGING ANAPHORA RESOLUTION FOR THE RUSSIAN LANGUAGE

Roitberg A. M. (cvi@yandex.ru)[1,2],
Khachko D. V. (mordol@lpm.org.ru)[1]

[1]IMPB RAS- Branch of KIAM RAS, Puschino, Russia;
[2]School of Linguistics HSE RSU, Moscow, Russia

Presented in this report are the initial findings of automatic bridging anaphora recognition and resolution for the Russian language. For a resolution of F-measure = 0.65 we use a manually-annotated bridging corpus and machine-learning techniques to develop a classifier to predict bridging anaphors, bridging anchors, and bridging pairs. In addition to this, we discuss the features used for the classifier and discuss the importance of each feature. Experimental results show that our classifier works well, however, potential improvements can be made, these improvements will be explored.

Key words: bridging anaphora, bridging anaphora recognition, bridging anaphora resolution

РАЗРЕШЕНИЕ БРИДЖИНГ АНАФОРЫ НА МАТЕРИАЛЕ РУССКОГО ЯЗЫКА

Ройтберг А. М. (cvi@yandex.ru)[1,2],
Хачко Д. В. (mordol@lpm.org.ru)[1]

[1]ИМПБ РАН—филиал ИПМ им. М. В. Келдыша РАН),
Пущино, Россия; [2]Школа лингвистики НИУ ВШЭ

В статье представлены первые результаты автоматического распознавания бриджинг-анафоры для русского языка. Для распознавания анафоры F-мера равна 0,65. Распознавание бриджинг-анафоры проводилось с применением методов машинного обучения. Для обучения классификатора мы использовали корпус коротких новостных текстов на русском языке с ручной разметкой бриджинг-анафоры. В данной работе обсуждаются наборы признаков, которые использовались для обучения, а также значимость каждого из признаков.

Ключевые слова: бриджинг-анафора, распознавание бриджинг-анафоры, разрешение бриджинг-анафоры

1. Introduction

Bridging anaphora, cover a broad class of semantic and discourse relations and play an important role in text cohesion. Therefore, bridging anaphora recognition and resolution is of vital importance for a variety of different NLP tasks.

Thus far, automatic bridging anaphora resolution has been in its early stages of development. Previous studies on bridging resolution include: [Poesio et al. 2004, Fan et al 2005] which uses a semantic approach to bridging resolution, they restrict bridging relations to several semantic relations, like part-of, set-subset etc. [How et al. 2013; Sasano 2009] considers using probabilistic methods, finally, in the last paper, bridging anaphora is resolved just as zero anaphora is. We too use a probabilistic approach for bridging anaphora recognition and resolution. Currently, there are no previous studies on bridging anaphora for the Russian language; our goal is to bridge this gap.

Following [Clark 1975], we define bridging as all anaphoric relations between non-coreferential entities. Clarks definition is very general; all the works on bridging anaphora narrow this definition in some way. For example; [Lüngen 2008; Poesio 2004] restrict bridging-anaphora semantically, [Hou et al. 2013] restricts all bridging expressions to noun phrases etc.

We apply a completely new syntactic type of restriction to bridging cases (See 2.1). Our non-semantic restriction approach is attributed to the fact that we don't apply WordNet or a similar resource for bridging resolution, as distinguished from [Poesio et al. 2004] for example. Unfortunately, there is no such resource as English WordNet for Russian.

The paper is set out as follows: In Section 2.1 we describe our syntactic-oriented approach—Genitive Bridging; Section 2.2 details the training and testing corpus used throughout the paper—RuGenBridge, Section 3 explores our machine-learning experiments, first describing in general terms the machine learning procedure (3.1), then going on to list the learning features for bridging elements (3.2.1), bridging anchors (3.2.2) and bridging pairs (3.2.3). In Section 4 we present the results and discuss the advantage of our methods and explore future directions of research.

2. Data collection and Preparation

2.1. Genitive Bridging

We applied a specific syntactic-oriented approach to bridging anaphora called 'genitive bridging'. We capture the bridging relation in cases where the anchor NP and the bridging-anaphor NP are: 1) anaphorically linked and 2) the heads of anaphorically linked NPs can form a grammatical genitive construction. The bridging anaphor is the head of genitive construction and the anchor a genitive dependence.

(1) *Tam stoyal <u>gruzovik</u> s naklejkami na* ***kabine****.*
'*There was a huge <u>truck</u> with stickers on the* ***cab***'

Example 1: *kabina—gruzovik* [*cab—truck*] is a case of genitive bridging: 1) the entities are anaphorically linked; the cab of the truck was mentioned in the previous sentence; 2) at the same time *kabina gruzovika [cab truck.Gen] 'cabin of the truck'* is a grammatical genitive construction in Russian. So, in such a case, we annotate with genitive bridging: *cab* → *truck*.

We don't restrict bridging relations to some semantic relations. However our observations show that most of the genitive bridging pairs are as follows: 1) part-of relations, as in the example above; 2) political positions—geographic name: *president—USA*; position—organization: principal—school; 3) something located somewhere: *schools—Moscow*; 4) object—its possessor: *flat—landlord*; 5) expressions with names of measures: *oil—barrel*); 6) collocations, mostly deverbative nouns: *rates—increase, robbery—bank*. For further details see [Roitberg, Nedoluzhko 2016]

Note that; among the features used for bridging recognition in [Poesio 2004-B] there is one that can be described in the following way: two expressions are more likely linked to a bridging relation if they frequently appear in syntactic construction *X of Y*. To evaluate it, one must investigate several potential google queries of the form "the NBD of the NPA", where NBD is a head noun of bridging description and NPA is a head noun of a potential bridging antecedent. X of Y is a standard translation for Russian genitive construction X + Y.Gen.

2.2. RuGenBridge Corpus

To train and test our bridging-recognition system, we annotated the Russian corpus, highlighting the genitive bridging—RuGenBridge. It consisted of short news texts (up to 10 sentences) from internet news sources. Currently, we have annotated 339 texts or 61,076 tokens, and have tagged 609 genitive bridging pairs.

Segments of speech and syntactical links were annotated automatically by FreeLing1 and MaltParser2 [Nivre et al 2006] respectively. Bridging relations were annotated manually using BRAT3 tool.

The first part of the corpus (190 texts) was manually annotated by two annotators, with the agreed F-measure = 0.7. The remainder of the text was annotated by 1 annotator.

We annotated genetive bridging relations and coreferential chains for bridging anaphors and anchors. See Example 2.

(2) *Posle vozvrasheniya iz Irana on rassakazal o poezdke v etu stranu. «V Irane ochen' gostepriimnyj **narod**»*
*'After his returning from Iran, he told about the journey to this country. «In Iran there are very welcoming **people**»'*

In Example 2 we annotated the bridging link *narod* → *Iran* ('*people* → *Iran*') and the coreferential chain *Iran—strana—Iran* '*Iran—country—Iran*'. We postulate bridging relations between the bridging anaphor and the whole anchor's coreferential chain, as in the Prague Dependency Bank [Poláková 2013]. We consider two annotations as equal if their bridging anaphors are identical and their anchors belong to the same coreferential chains.

As well as the manually-annotated corpus, we also used a 5 million automatically-part-of-speech-tagged news corpus to train the Word2Vec4 model. Later we use

[1] http://nlp.lsi.upc.edu/freeling/

[2] www.maltparser.org

[3] http://brat.nlplab.org

[4] https://code.google.com/archive/p/word2vec/

Word2Vec outputs to calculate semantic similarity measures between the nouns of the texts and bridging anaphors or anchors that have been manually annotated.

3. Machine Learning Techniques for Bridging Resolution

For machine learning experiments, we use Python with libraries Pandas5 and Scikit-learn tool6. To reveal cases of bridging anaphora, we use a Random Forest Classifier algorithm because it produces the highest quality results, however, we also conducted some experiments with Logistic Regression and Decision Tree algorithms (see Section 4).

3.1. Procedure

Firstly, we use the whole corpus to calculate TF-IDF for all words in the corpus. Once this is done, we divide our corpus into two unequal parts and use two-step machine learning procedures to train our classifier (See 3.1.1, 3.1.2). We use the larger part (80,000 tokens) called Part 1 bellow for Step 1 and the smaller (14,000 tokens)—Part 2 bellow for Step 2. Step 1 involves training Classifier 1 to recognize potential bridging anaphor/anchors; Step 2 involves training Classifier 2 to recognize bridging pairs. For both steps we apply cross-validation techniques with k-fold = 4. The average was then calculated with the AUC measure also being calculated after each run.

3.1.1. First Part: Step 1

We take all of the bridging anaphors/anchors from Part 1 of the corpus and choose the 10 most semantically similar nouns for each bridging anaphor/anchor (according to Word2Vec).

Once this is done we train the classifier to predict bridging anaphors/anchors. We use an analogous procedure for bridging anaphors and anchors. Let us consider anchors for example: we take all manually annotated anchors as positive examples and add to this “positive” set a group of random nouns as negative examples. The negative set is seven times larger; the best proportion between positive and negative examples was experimentally derived. This data is used for Classifier 1. This classifier was then used to predict bridging anaphors/anchors in the second step.

3.1.2. Second Part: Step 2

As previously mentioned, we use the first step classifier to automatically annotate bridging anaphors and anchors in Part 2. For this task, we optimize Classifier 1 to a very high precision ($P = 0.98 - 1.00$), with such settings it identifies almost all bridging elements/anaphors, moreover, it identifies 10 times more wrong nouns. We then take all of these bridging elements and anchors, match them to Golden standard, and mark the real bridging pairs as positive examples and wrong bridging pairs as negative examples. Finally, we use this data to train Classifier 2 to predict bridging pairs.

[5] http://pandas.pydata.org

[6] http://scikit-learn.org/stable/

3.2. Features

We use 2 different feature sets for the bridging anaphors/anchors classifier (Classifier 1) and bridging-pairs classifier (Classifier 2).

3.2.1. Step 1. Feachors for Anchors Prediction

We used eight features to train a classifier for anchor recognition and prediction. These features include:

1. *Semantic similarity to anchor anaphor*—as previously mentioned, we took the 10 most similar words to each bridging anaphor/anchor, in order to determine whether the word is in the list of comparable words to bridging anaphors/anchors according to Word2vec data.
2. *TF-IDF of word*—TF-IDF measure shows how important a word is to a document in a collection or corpus. This feature highlights the tendency anchors have to being in given information.
3. *Linear Distance*—the distance from the beginning of the text to the anchor, calculated in words.
4. *Lemma*—is it a lemma match to one of the anchors lemmas annotated in first part of the corpus?
5. *Type of syntactic link from the NP's head to word*—The MaltParser syntactic link type from the head of this word to the word. MaltParser uses a set of syntactic relations developed for SyntagRus [Boguslavsky et al 2006].
6. *Case*—a case automatically tagged by FreeLing.
7. *Syntax distance*—the shortest way from the bridging anaphor to the sentence root in the dependency tree is automatically built by MaltParser.
8. *Animacy*—an animacy or inanimacy automatically tagged by FreeLing.

3.2.2. Step 1. Feachors for Bridging Anaphors Prediction

For bridging anaphor recognition, we used the same number of features as the anchor feature set. The Classifier tends to identify nearly 70% of all nouns in the text as potential bridging anaphors. This is one of the reasons that there are no articles in Russian. On the other hand, alternative markers of definite NPs, such as deictic and possessive pronouns, seem to have a narrower distribution in Russian than in Romanic and Germanic languages where definite NPs are usually considered typical bridging anaphors. The fact that anaphors are generally less specific than antecedents, this can reflect on bridging anaphor recognition. These considerations need further exploration.

3.2.3. Step 1. Feachors for Bridging Pairs Prediction

The features used to train the classifier for bridging-pairs recognition include:

1. *Linear Distance*—linear distance between the bridging anaphor and the anchor.
2. *Probability of anchor*—estimated probability of the potential anchor computed by our first step classifier.
3. *Probability of bridging anaphor*—the estimated probability of the potential bridging anaphor computed by our first step classifier.
4. *Lemma of bridging anaphor*—is it a lemma match to one of the bridging anaphors lemmas annotated in first part of the corpus?

5. *Syntactic distance*—the shortest way from the bridging anaphor to the sentence root plus the shortest way from the anchor to the sentence root; if the bridging anaphor and anchor are in different sentences we just add 2, because we consider the texts a main root.
6. *Lemma of anchor*—is it a lemma match to one of the anchors lemmas annotated in first part of the corpus?
7. *Case of Bridging anaphor*—a case of the potential bridging anaphor automatically tagged by FreeLing
8. *Case of anchor*—a case of the potential anchor automatically tagged by FreeLing.

4. Experimental Results

For all experiments, we used cross-validation techniques for training, and an AUC measure for evaluating results. AUC is a square under the Receiver Operating Characteristic (ROC) curve. The ROC curve shows a correlation between the true positive rate (TPR) and the false positive rate (FPR) as seen in the graphs in section 4.3. An advantage of this measure is discussed in [Ling 2003]. AUC measure is a common measure for machine-learning experiments and classifier evaluation. The F-measure was also determined so that we could compare our results to related studies.

4.1. Machine-learning Algorithms

At the start of the study, we applied different machine learning algorithms to train the classifier. Three algorithms that are considered to be the least sensitive to correlated features that are common in natural language data are: Random Forest, Logistic Regression, and Decision Tree. For this study we chose Random Forest, which produced the most reliable results. For each algorithm, we tried different options, but such technical details are beyond the scope of this paper. The AUC measure for predicting anchors, bridging anaphors, and bridging pairs are given in Table 1.

Table 1. The application of different machine-learning algorithms results

	Random Forest	**Logistic Regression**	**Decision Tree**
AUC—Anchors	0.981	0.94	0.85
AUC—Bridging anaphors	0.969	0.93	0.92
AUC—Bridging pairs	0.92	0.79	0.7

4.2. Feature Importance

The semantic similarity feature is the most important feature of our set, followed by TF-IDF. Other features are less important, but all the features working in conjunction provide significant improvement.

Table 2. Anchor's features and features' importance

Anchor's Feature	Feature Importance	AUC without Feature
Semantic similarity to anchor/bridging anaphor	0.54	0.85
TF-IDF of word	0.17	0.98
Distance in words from the beginning of the text	0.07	0.98
Lemma	0.06	0.97
Type of syntactic link from the NP's head to the word	0.03	0.97
Case	0.02	0.98
Syntax distance from word to root of sentence	0.02	0.98
Animacy	0.01	0.97

Table 3. Bridging pairs' features and feature contributions

Bridging Pair's Feature	Feature Importance	AUC without Feature
Distance from bridging anaphor to anchor in words	0.25	0.82
Probability of anchor	0.23	0.83
Probability of bridging-anaphor	0.19	0.84
Lemma of bridging-anaphor	0.09	0.85
Syntactic distance from bridging anaphor to anchor	0.08	0.85
Lemma of anchor	0.05	0.88
Case of bridging anaphor	0.04	0.87
Case of anchor	0.03	0.88

4.3. Bridging Resolution Results

Our trained classifier shows strong results in anchor recognition (AUC=0.981) and weaker results for bridging anaphor recognition (AUC = 0.969). Poor bridging anaphor recognition impacts bridging pairs recognition in turn and for bridging pairs AUC=0.92. This result is not as high as we hoped it would be, but it is a satisfactory preliminary result and we believe it can be improved by extending the corpus and optimizing feature sets.

All the results obtained are presented in the charts below (Fig. 1–3). Included; are (ROC) curves. The ROC curve shows a correlation between the true positive rate (TPR) and the false positive rate (FPR). The AUC measure is the Area Under the Curve. For result "by chance" AUC=0.5

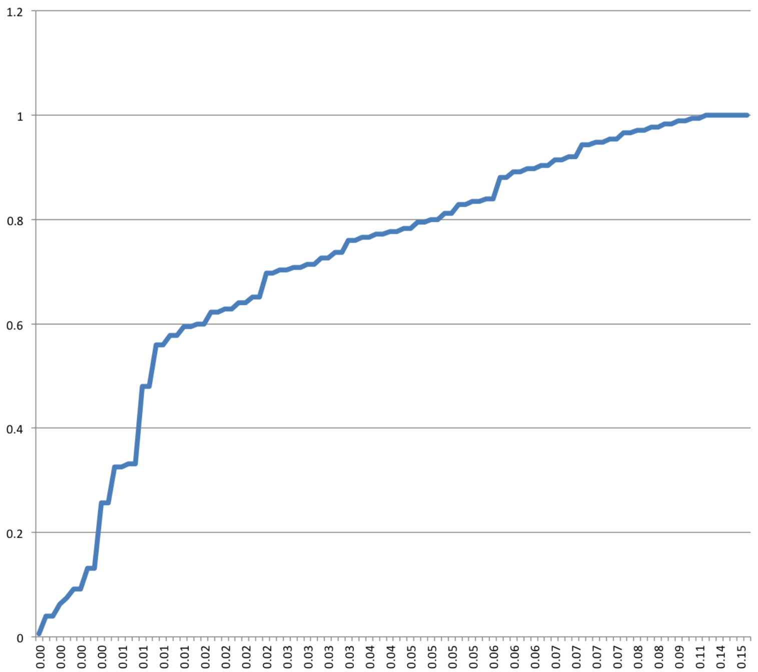


Fig. 1. ROC for Anchors. TPR is vertical, FPR is horizontal, AUC = 0.981

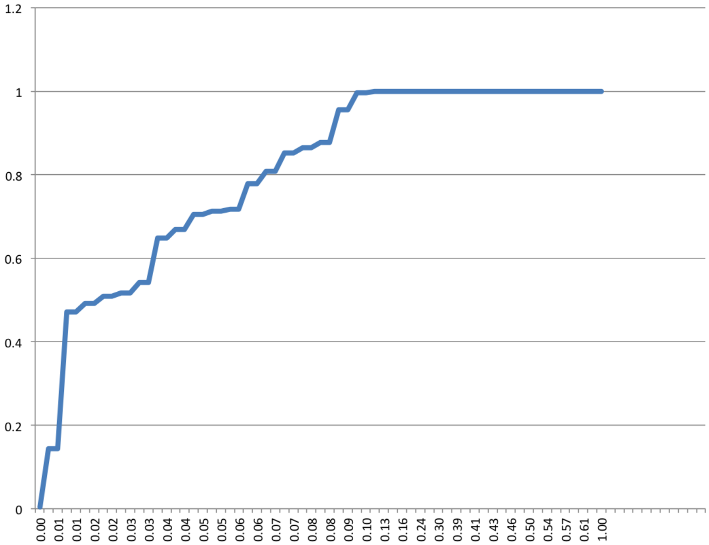


Figure 2. ROC for Bridging Anaphors. TPR is vertical, FPR is horizontal. AUC = 0.969

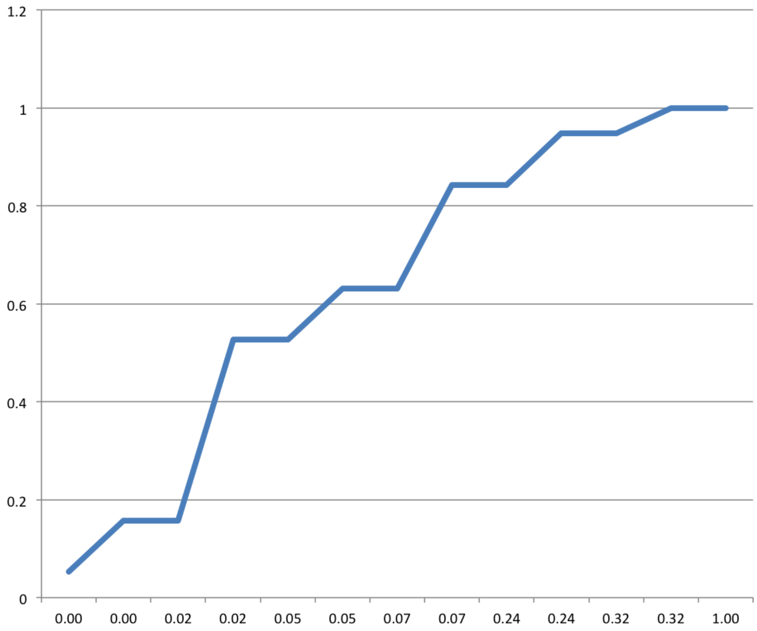


Figure 3. ROC for Bridging Pairs. TPR is vertical, FPR is horizontal. AUC = 0.92

F-measure was determined for bridging-anaphor and anchor recognition and bridging resolution. We can vary precision, recall and F-measure values by assigning the value of confidence score—the score shows the level of system confidence of the bridging element, anchor or pair. For bridging elements and anchors we vary the value of the rate in order to maximize precision. For bridging pairs we try to maximize F-measure. Maximum F-measure corresponds to a confidence rate of 0.35.

Results are presented in the tables bellow:

Table 4. Precision, recall and F-measure for bridging elements and anchors

	Bridging Element	**Anchors**
Precision	1	0.98
Recall	0.21	0.20
F-measure	**0.35**	**0.61**

The Table below shows the results for our bridging resolution system:

Table 5. Precision, recall and F-measure for bridging pairs

Precision	0.58
Recall	0.73
F-measure	**0.65**

4.4. Dependency Between Corpus Size and Bridging Recognition Quality

In Figures 4–5, the correlation between corpus size and classifier results quality is shown. The red line is the AUC computed for 100% of our data, 90% of our data, 80% and so on. Vertical intervals are the AUC measure dispersion between different runs of the classifier, which were trained on the mentioned corpus size. We have provided 10 runs for each variant of corpus size, from 100% to 40%. We changed the corpus size with 10% intervals.

The growth of the curves while using almost 60% of the data set for the anchor chart and close to 50% of the data set for bridging pairs was not the typical growth usually seen. The function should decrease monotonically from 100% to 40–50%. These growths could be due to a variety of specific text in training data.

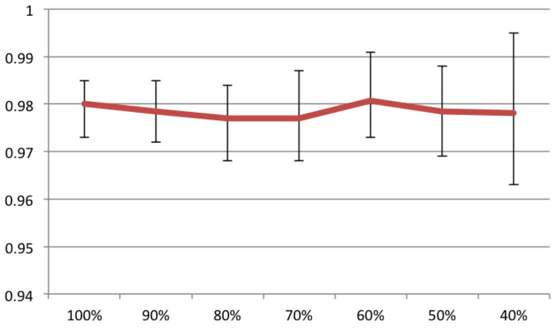


Figure 4. Dependency between corpus size and anchor recognition

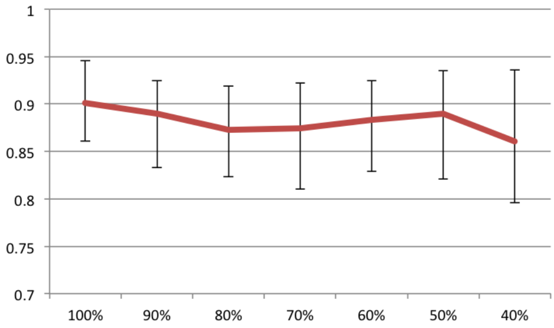


Figure 5. Dependency between corpus size and bridging-pairs recognition

5. Discussion

Automatic bridging resolution and recognition is still in its early stages of development. All other projects use different approaches to the sets of restrictions to bridging, so it's difficult to compare our results with other bridging resolution and recognition studies. We found that our results (F-measure = 0.65) are high enough for such a complicated task, when compared to the F-measure used for the bridging resolution system for German [Klenner, Manfred, et al.], which is varied between 0.58 to 0.61.

Despite the preliminary results being adequate, we are going to continue our work with the goal of improving the level of automatic bridging resolution. Firstly, we plan to extend our corpus and optimize a feature set, extending the corpus should increase the result of bridging resolution. As shown in the extension in Figures 4–5, we have not yet reached a plateau, where increasing the data does not greatly influence the results. In relation to the features, we want to first improve the quality of syntactic features. For instance, currently we use all syntactic link types provided by MaltParces to compute the "type of syntactic link" feature while training the classifier; MaltParser distinguished more than 60 types of syntactic links. It is apparent that dividing all these syntactic link types into several groups so that the feature will have ten times less values results in a more effective feature. Also, we want to add a feature: "in one sentence" for bridging pairs, we expect that this will balance the "syntactic distance" feature, which is better than just adding 2, in the case of the bridging anaphor and anchor being in different sentences. There are also other features that are currently being considered for implementation.

Our approach for bridging resolution is simple. There is no need to use complicated, pre-prepared resources such as WordNet; these are only accessible for a small number of languages. To train a classifier, one simply needs a small, manually-annotated, corpus and automatic-annotation tools, which are well developed for a multitude of languages. Therefore, we hypothesize that our method can be applied to different types of bridging and different languages.

This work was supported by grants 16-04-01640 A and 15-07-09306 from RFBR.

References

1. *Boguslavsky, I., Grigorieva, S., Grigoriev, N., Kreidlin, L., & Frid, N.* (2000, July). Dependency treebank for Russian: Concept, tools, types of information. In Proceedings of the 18th conference on Computational linguistics-Volume 2. Association for Computational Linguistics, pp. 987–991
2. *Clark, H. H.* (1975, June). Bridging. In Proceedings of the 1975 workshop on Theoretical issues in natural language processing, Association for Computational Linguistics, pp. 169–174
3. *Fan, J., Barker, K., & Porter, B.* (2005, October). Indirect anaphora resolution as semantic path search. In Proceedings of the 3rd international conference on Knowledge capture ACM, pp. 153–160
4. *Hou, Y., Markert, K., & Strube, M.* (2013). Global Inference for Bridging Anaphora Resolution. In HLT-NAACL pp. 907–917
5. *Klenner, M., Tuggener, D., Fahrni, A., & Sennrich, R.* (2010, August). Anaphora resolution with real preprocessing. In International Conference on Natural Language Processing, Springer Berlin Heidelberg pp. 215–225
6. *Lassalle, E., & Denis, P.* (2011, October). Leveraging different meronym discovery methods for bridging resolution in French. In Discourse Anaphora and Anaphor Resolution Colloquium, Springer Berlin Heidelberg, pp. 35–46

7. *Ling, C. X., Huang, J., & Zhang, H.* (2003, June). AUC: a better measure than accuracy in comparing learning algorithms. In Conference of the Canadian Society for Computational Studies of Intelligence (pp. 329–341). Springer Berlin Heidelberg.
8. *Lüngen, H.* (2008). RRSet-Taxonomy of rhetorical relations in SemDok. Interne Reports der DFG-Forschergruppe, 437
9. *Nivre, J., Hall, J., & Nilsson, J.* (2006, May). Maltparser: A data-driven parser-generator for dependency parsing. In Proceedings of LREC , Vol. 6, pp. 2216–2219
10. *Poesio, M., Delmonte, R., Bristot, A., Chiran, L., & Tonelli, S.* (2004-A). The VENEX corpus of anaphora and deixis in spoken and written Italian. University of Essex.
11. *Poesio, M., Mehta, R., Maroudas, A., & Hitzeman, J.* (2004-B, July). Learning to resolve bridging references. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, p. 143
12. *Poláková, L., Mírovský, J., Nedoluzhko, A., Jínová, P., Zikánová, S., & Hajicová, E.* (2013). Introducing the Prague Discourse Treebank 1.0. In IJCNLP (pp. 91–99).
13. *Roitberg, A., & Nedoluzhko, A.* (2016). Bridging Corpus for Russian in comparison with Czech. Coreference Resolution beyond OntoNotes, p. 59.
14. *Sasano, R., & Kurohashi, S.* (2009, August). A probabilistic model for associative anaphora resolution. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, Association for Computational Linguistics, pp. 1455–1464

EXPLOITING RUSSIAN WORD EMBEDDINGS FOR AUTOMATED GRAMMEME PREDICTION

Romanov A. V. (Aleksey_Ro@abbyy.com)

ABBYY; Moscow Institute of Physics and Technology (MIPT), Moscow, Russia

Distributed representations of words are currently used in a variety of linguistic tasks. A specific branch of their possible applications includes automatic extraction of word-level grammatical information by formulating it as a problem of word embedding classification. In this paper, we investigate applicability of this approach to prediction of several particular classifying grammemes. We focus on animacy of Russian nouns and transitivity of Russian verbs. These categories can serve as good examples of classifying grammatical categories in the Russian language since their concrete values can hardly be predicted judging by appearance of words and morphemes that constitute them. We conduct experiments on a corpus of Russian texts from the Web with several widely used word-embedding algorithms and different parameter settings. Experimental evaluation includes the comparison of performance of several classifiers, with distributed representations being source of features for classification task. Our findings show feasibility of the approach and its potential to be implemented for solving related tasks.

Key words: natural language processing, distributional semantics, word embeddings, word-level classification, automatic corpus annotation

ПРИМЕНЕНИЕ МОДЕЛЕЙ ДИСТРИБУТИВНОЙ СЕМАНТИКИ ДЛЯ АВТОМАТИЧЕСКОГО ПРЕДСКАЗАНИЯ ГРАММЕМ

Романов А. В. (Aleksey_Ro@abbyy.com),

ABBYY; Московский физико-технический институт (государственный университет), Москва, Россия

1. Introduction

Distributed word representations, or word embeddings, have already shown their power as a basis for efficient model training within the scope of neural-network approach in various natural language processing tasks. In addition to this primary mission, word embeddings are widely and successfully applied to development of solutions that do not necessarily use neural nets as their core component; e.g. contextual information encoded in the distributed representations can be used for word similarity estimation and in related problems.

Another potential application of word embeddings resides in automated word-level grammatical and semantical information extraction. This set of tasks is itself quite interesting for linguists: measuring the correlation between contexts of the word and its internal sense, and determining the limits of distributional approach are two questions that are still open and should be investigated broader. Moreover, such tasks can be seen as auxiliary for more complex ones. One can consider, for example, the following situation: vast amount of text available on the Web can be exploited in a variety of linguistic studies provided it is properly and fully labeled in accordance with the specific task orientation. Availability of automated word labeling methods for text corpora is thus the condition for future linguistic research.

In this paper, we investigate applicability of distributed word representations to prediction of several classifying grammemes of Russian words. In particular, we consider animacy of nouns and transitivity of verbs in the Russian language, since concrete values of these grammatical categories can hardly be predicted judging by appearance of words and morphemes that constitute them. We expect that good performance of automatic classification in the aforementioned tasks may open the way to extension of the approach into other related problems.

We propose to utilize real-valued vectors obtained from distributed representation models as features in these prediction tasks, which may lead to a scheme of grammeme prediction on a basis of insufficiently labeled corpora. We explore power of several widely used word-embedding algorithms and train models with different sets of parameters in order to achieve better performance. Additional investigation concerns testing the dimensionality reduction technique proposed recently ([12]) for enhancing word embeddings in applied tasks. Based on the experimental results, we argue that our approach is feasible for prediction of grammatical characteristics of Russian words.

2. Related work

The task of automated grammeme prediction is closely related to a more general problem of automated corpus annotation and, more specifically, to automated grammatical tagging. A classic work in the field is [9], where the authors utilize a stochastic algorithm to complete the first stage of two-staged tagging process, the second being manual correction of errors produced by the automatic stage. The method and a number of related ones ([13], [7]) are based on complex models with a large number of parameters to be tuned in order to achieve good performance; this may be an encumbrance in the case of small corpus on annotated data.

State-of-the-art techniques of automatic grammatical tagging mostly focus on overcoming the obstacle of insufficient amount of data available for model training. This is important, among other things, for developing automated tagging systems for languages lacking high-quality text resources and corpora on the Web. The authors of [6] propose to use graph-based label propagation for cross-lingual knowledge transfer and utilize the resulting labels as features in an unsupervised model. The idea is further developed in [14], where ambiguous learning approach enables effective automated transfer of tags from English corpora to corpora in other languages. In [8] several techniques for low-resource tagging are shown to be feasible.

Word embeddings have also been utilized for solving a number of morphological tasks. A work [4] proposes an architecture and an algorithm performing well in POS-tagging task without labeling data beforehand. [5] describes a model of morphologically guided embeddings, which is capable of handling tagging tasks in a semi-supervised manner by adding labeling to the training corpus.

Several works are devoted to automatic animacy prediction ([3], [1]). The methods rely on hand-crafted features obtained from annotated sources of semantic and lexical information, achieving high accuracy over 90%. The key idea of our method is, in contrast, in taking automated grammeme prediction to a competitive level by means of minimal available corpus annotation and limited amount of data. Our approach builds upon and extends the method described in [12], where the authors explore the ability of word embeddings to predict certain grammatical functions including noun animacy. In our work, we try to extend their research into the Russian language and improve the performance studying the influence of various parameters, both on the stage of distributional model training and while generating classification features.

3. Method description

3.1. Formal problem statement

We consider the task of grammeme prediction as a word-level binary classification task. In other words, we train a classifier to predict whether a word has a certain value of a grammatical category or not. In our study, which is designed to give preliminary characterization to feasibility of the approach, we do not focus on homonym disambiguation and treat each unique sequence of characters as a classification object.

Thus, the task is to build is a classifier $a\colon W \longrightarrow \{0,1\}$ that, on the basis of feature representation $\boldsymbol{w}$ of the word $w \in W$, would predict whether w has the grammeme g (1 class) or not (0 class). The optimal classifier a^* is found by training on the set of labeled precedents $W = \{(\boldsymbol{w}_1, y_1), \dots, (\boldsymbol{w}_n, y_n)\}$, $y_1 \in \{0, 1\}$, $i = 1 \dots n$, i.e. the process of minimization of the empirical risk $Q(a, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}[a(\boldsymbol{w_i}) \neq y_i]$:

$$a^* = \arg\min_{a} Q(a, D)$$

3.2. Grammeme choice

In order to verify the assumption that word embeddings may potentially be applied for grammeme prediction, we have chosen two of *classifying* grammemes in the Russian language, i.e. those that are intrinsically fixed for a lexeme and constant across its derived forms. Noun animacy and verb transitivity are good examples of grammatical categories whose values cannot be easily predicted judging by appearance of words and morphemes that constitute them; therefore, it is particularly interesting if distributional models of morphologically rich languages, with Russian being an example, can be a source of ready-to-use classification features.

Noun animacy basically provides distinction between nouns referring to humans (and some other biological creatures) and those referring to various inanimate objects and phenomena. In the Russian language, it is often necessary to have information about the noun animacy in order to inflect the noun correctly. Consider, for example, the plural accusative of an animate noun *мальчик* ("boy") — *мальчиков*, matching the plural genitive, and the plural accusative of an inanimate noun *пальчик* ("little finger") — *пальчики*, matching the plural nominative. This rule generalizes to other animate and inanimate nouns. Automated animacy/inanimacy prediction is thus useful for morphological analysis and phrase generation as well.

Verb transitivity is a property of a verb to take direct objects, a special case of a more general notion of valency. In Russian, like in English, this category is expressed syntactically, i.e. it is possible to identify an intransitive verb by attempting to supply it with an appropriate direct object but not by judging by its morphological markers. Transitivity used to be believed to be a binary characteristics of a verb; now, no verbs are mainly seen as "absolutely transitive" but rather "more often occurring in texts with a transitive role". Intransitive verbs, however, never appear in phrases with direct objects, and this fact enables the task of transitivity prediction to be considered as a binary classification task.

3.3. Features

The main idea of the method is to use pre-trained word embeddings "as is" as features for classification. The advantages of this approach are its simplicity and scalability onto related problems. We tested different parameter configuration sets of distributional model training to study the effect of the configuration choice on the overall performance.

Additional experiments were devoted to:

- enriching feature space with auxiliary per-word information provided in distributional models;
- transformations of word embeddings aimed at obtaining more informative representations.

4. Experiments

4.1. Text Data

Distributional models were trained on a collection of Wikipedia articles in the Russian language (1.3M articles and 100M tokens on the whole). The text was split into sentences and lowercased. Non-Cyrillic words and punctuation marks were removed. All digits and numbers were replaced by a single special token. We lemmatized the corpus with *pymorphy2*, a Python package.

A list of Russian nouns and verbs labeled respectively with animacy and transitivity tags was obtained from the *pymorphy2* package as well. Overall, 12K verb (7.5K transitive) and 121K noun (47K animated ones, including proper names) lemmas were extracted and prepared for classification.

4.2. Distributional model training

Among frameworks offering opportunities of training distributional models, *gensim* and *fasttext* were chosen, with word2vec and FastText being the models providing word embeddings.

Word2vec continuous-bag-of-words [10] models were trained with a set of default parameters. We tried different (symmetric and asymmetric) configurations of context windows in order to test a hypothesis that smaller context windows induce word embeddings with greater grammeme prediction power. We also varied the dimension of embeddings, as higher dimension leads to better performance in a number of related tasks.

FastText model [2] is a promising extension of word2vec, designed to construct vectors not only for words but also for character N-grams that constitute them. This way, the words that have some N-grams in common get representations that are more similar to each other. Another useful feature of this approach is its ability to predict vectors for unseen words. FastText models of various dimensions were trained as well.

4.3. Experimental results

In our experiments, we compared three types of classifiers: Support Vector Machine (SVM), Random Forest (RF) and Multi-Layer Perceptron (MLP). The metrics to be measured was weighted F1 score (average F1 by classes weighted by support). Hyperparameters of classifiers (regularization constants, number of trees and hidden layers, respectively) were tuned to obtain the best performance on 5-fold stratified cross-validation scheme.

Window size and dimension effect

We selected several word2vec and FastText models to study the effect of different training parameters on classification performance (i.e. on weighted F1 scores):

- word2vec, 250-dimensional vectors, context window: 5 words before + 5 words after the word;
- word2vec, 500-dim, context: 5 + 5;
- word2vec, 500-dim, context: 2 + 2;
- word2vec, 500-dim, context: 0 + 3;
- word2vec, 500-dim, context: 3 + 0;
- FastText, 250-dim, context: 5 + 5;
- FastText, 500-dim, context: 5 + 5;
- FastText, 250-dim, context: 5 + 5, prediction of vectors for unknown words;
- FastText, 500-dim, context: 5 +5, prediction of vectors for unknown words.

It is worth noting that in the last two cases, the support for classification task is much greater in size than that of other cases: in such a setting we can obtain vectors for all the words we are willing. On the contrary, in the first cases, we conduct classification on the set of words that occur in the Wikipedia corpus. Thus, in the last two cases we have 121K nouns (54K in the first cases) and 12K verbs (6.4K in the first cases) for classification. Therefore, classification performance may differ in these two cases, but it resembles the situation of automated corpus tagging to a greater extent. The results are given in Table 1.

Table 1. Performance of different distributional models and classifiers

Model	Transitivity			Animacy		
	SVM	RF	MLP	SVM	RF	MLP
word2vec, 250, 5+5	0.818	0.767	0.810	0.880	0.877	0.871
word2vec, 500, 5+5	0.833	0.748	0.831	**0.888**	0.870	0.873
word2vec, 500, 2+2	0.657	0.556	0.644	0.659	0.653	0.626
word2vec, 500, 3+0	0.628	0.550	0.624	0.634	0.621	0.601
word2vec, 500, 0+3	0.631	0.558	0.620	0.631	0.615	0.612
FastText, 250	0.853	0.827	0.862	0.845	0.834	0.825
FastText, 500	0.859	0.834	**0.868**	0.848	0.820	0.830
FastText, 250, prediction	0.828	0.819	0.856	0.790	0.799	0.805
FastText, 500, prediction	0.840	0.825	0.862	0.797	0.789	0.811

The results show that the models that incorporate knowledge about character N-grams of the word are more powerful for transitivity prediction, and their features have non-linear dependencies. However, linear methods performed better on classic word2vec models for animacy prediction task, since noun animacy in the Russian language has weak correlations with the words' appearance. Overall, the results are comparable with those achieved in [12] for tasks in other languages.

Enriching embeddings with auxiliary training information

While training word2vec models, one can access both W_{in} and W_{out} matrices. The former is mainly used as the word embedding source, and the latter rarely comes into use in practical tasks. We investigated the applicability of both types of vectors to our problem in the following manner. Three sets of classification features were extracted from a 500-dimensional word2vec model with the default 5+5 context window:

- W_{in} rows—as in the abovementioned experiment;
- W_{out} columns;
- stacked W_{in} rows and W_{out} columns.

Table 2 shows the results of this study.

Table 2. Performance on various matrix-based features

Features	Transitivity			Animacy		
	SVM	RF	MLP	SVM	RF	MLP
W_{in} rows	0.833	0.748	0.831	0.888	0.870	0.873
W_{out} columns	0.848	0.755	0.844	0.886	0.871	0.865
stacked $W_{in} + W_{out}$	**0.852**	0.750	0.841	**0.893**	0.876	0.883

Reducing word embeddings by the main PCA components

We applied the trick proposed in [11], which is said to create more powerful embeddings performing better in a bunch of tasks. The idea of the trick is to center embeddings, reducing them by their average vector, apply PCA and remove *D* most informative components from the vectors afterwards. We studied classification performance with several values of *D* on the same distributional model as in the experiment described in the previous paragraph. The results are available in Table 3.

Table 3. Effect of reduction by PCA components on overall performance

D	**Transitivity**			**Animacy**		
	SVM	**RF**	**MLP**	**SVM**	**RF**	**MLP**
—	**0.833**	0.748	0.831	**0.888**	0.870	0.873
2	0.822	0.735	0.816	0.795	0.822	0.864
3	0.826	0.733	0.822	0.787	0.818	0.862
5	0.821	0.737	0.811	0.739	0.782	0.851
10	0.815	0.720	0.811	0.729	0.749	0.839

4.4. Discussion

Performance achieved in the experiments is sufficient to claim feasibility of the approach. Surprisingly, smaller values of context window size used in word2vec training lead to significant drop in classification performance. This effect allows to assume that grammemes of animacy and transitivity are more closely related to broader, semantic contexts of a word than to narrower, syntactic ones.

FastText showed better performance than word2vec did in transitivity prediction task. This can be attributed to the fact that some transitive (or, probably, intransitive) verbs in the Russian language share certain character N-grams. Thus, a model that assigns closer vectors to similarly looking words is supposed to perform better in this task. At the same time, the problem of animacy prediction is solved better by word2vec models, since animate nouns cannot be distinguished from inanimate ones judging by their appearance. However, FastText predictive power for unseen words still makes it a good choice for automated corpus annotation.

It is worth noting that in some of the task settings non-linear classification models did not achieve higher performance than linear ones (especially on word2vec vectors). Another interesting fact is that W_{out} matrix of word2vec models is sometimes even more informative in classification tasks than W_{in} containing input word embeddings. Removing main PCA components did not drop significantly the quality of classification, but there was no increase as well.

We have also analyzed errors produced by classifiers in both tasks and can group them into the following categories:

- polysemantic words (e.g. изменить (transitive “to change” or intransitive “to cuckold”)) and homonyms (e.g. везти (transitive “to carry” or intransitive “to be lucky”, *барак* (“a barrack”) and Барак (“Barack”));

- rare words lacking occurrences in the training corpus (e.g. дефилировать “to sashay”, хлебопашец “a sodbuster”);
- transitive verbs frequently used in sentences without a direct object (e.g. петь “to sing”);
- inanimate proper names that can be seen as human names (e.g. Бредфорд “Bradford”).

Overall, the majority of classifying errors appear to arise due to labeling problems and, to a lesser degree, due to the limited amount of data for training the distributional model.

5. Conclusion

We propose a method of automatic grammeme prediction, which is based on word embedding classification and does not rely on corpora annotation. Preliminary findings show performance that is competitive with systems developed on handcrafted features.

As the future work, we plan to achieve better classification quality and extend the method to handle other grammatical categories than those described in this paper. Improvements can be made by preparatory homonym disambiguation or training on unlemmatized text with subsequent pooling on word forms for lemmas (which can be done by several promising schemes) during the classification stage. Another interesting course of future study includes extension of our approach to other languages.

Acknowledgements

We are grateful to Eugene Indenbom and Konstantin Zuev for their valuable comments and suggestions.

References

1. *Bloem J., Bouma G.* (2013), Automatic animacy classification for Dutch, Computational Linguistics in the Netherlands Journal, Vol. 3, pp. 82–102.
2. *Bojanowski P., Grave E., Joulin A., Mikolov T.* (2016), Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606.
3. *Bowman S. R., Chopra H.* (2012), Automatic animacy classification, Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, Montreal, pp. 7–10.
4. *Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P.* (2011), Natural language processing (almost) from scratch, Journal of Machine Learning Research, Vol. 12, pp. 2493–2537.
5. *Cotterell R., Schütze H.* (2015), Morphological Word-Embeddings, HLT-NAACL, pp. 1287–1292.

6. *Das D., Petrov S.* (2011), Unsupervised part-of-speech tagging with bilingual graph-based projections, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Portland, pp. 600–609.
7. *Federici S., Pirrelli V.* (1994), Context-sensitivity and linguistic structure in analogy-based parallel networks, Current Issues in Mathematical Linguistics, pp. 353–362.
8. *Garrette D., Baldridge J.* (2013), Learning a Part-of-Speech Tagger from Two Hours of Annotation, HLT-NAACL, pp. 138–147.
9. *Marcus M. P., Marcinkiewicz M. A., Santorini B.* (1993), Building a large annotated corpus of English: The Penn Treebank, Computational linguistics, Vol. 19(2), pp. 313–330.
10. *Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J.* (2013), Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, pp. 3111–3119.
11. *Mu J., Bhat S., Viswanath P.* (2017), All-but-the-Top: Simple and Effective Post-processing for Word Representations, arXiv preprint arXiv:1702.01417.
12. *Qiu P. Q. X., Huang X.* (2016), Investigating language universal and specific properties in word embeddings.
13. *Schmid H.* (2013), Probabilistic part-of-speech tagging using decision trees, New methods in language processing, Routledge, p.154.
14. *Wisniewski G., Pécheux N., Gahbiche-Braham S., Yvon F.* (2014), Cross-Lingual Part-of-Speech Tagging through Ambiguous Learning, EMNLP, Vol. 14, pp. 1779–1785.

RESEARCH OF A DEEP LEARNING NEURAL NETWORK EFFECTIVENESS FOR A MORPHOLOGICAL PARSER OF RUSSIAN LANGUAGE

Sboev A. G. (sag111@mail.ru)[1,2,3],
Gudovskikh D. V. (dvgudovskikh@gmail.com)[1],
Ivanov I. (honala@yandex.ru)[3],
Moloshnikov I. A. (ivan-rus@yandex.ru)[1],
Rybka R. B. (rybkarb@gmail.com)[1],
Voronina I. (irina.voronina@gmail.com)[4]

[1]National Research Center «Kurchatov Institute», Moscow, Russia;
[2]National Research Nuclear University «MEPhI», Moscow, Russia;
[3]Moscow Technological University (MIREA), Moscow, Russia;
[4]Voronezh State University, Voronezh, Russian Federation

In this study we present the method of morphological tagging on base of a deep learning neural network. The method includes two levels of an input sentence processing: individual characters level and word level. The comparison with other morphological analyzers was carried out with SynTagRus dataset in its original format of morphological characters, and its versions in Universal Dependencies formats 1.3 and 1.4. Achieved accuracies of Part-of-speech tagging: 98.34%, 98.49%, 97.60% (accordingly to each dataset). Results are a bit higher than the Google Syntaxnet accuracies and higher than the accuracies of the systems based only on Bidirectional Long short-term memory models. At the MorphoRuEval competition the method gained the third place.

Keywords: artificial neural networks, natural language processing, morphological parsing, PoS-tagging

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ В ЗАДАЧЕ МОРФОЛОГИЧЕСКОГО РАЗБОРА РУССКОГО ЯЗЫКА

Сбоев А. Г. (sag111@mail.ru)[1,2,3],
Гудовских Д. В. (dvgudovskikh@gmail.com)[1],
Иванов И. (honala@yandex.ru)[3],
Молошников И. А. (ivan-rus@yandex.ru)[1],
Рыбка Р. Б. (rybkarb@gmail.com)[1],
Воронина И. (irina.voronina@gmail.com)[4]

[1]НИЦ«Курчатовский Институт», Москва, Россия;
[2]НИЯУ «МИФИ», Москва, Россия;
[3]Московский Технологический Университет «МИРЭА», Москва, Россия;
[4]Воронежский Государственный Университет, Воронеж, Россия

1. Introduction

Nowadays there is a tendency to apply deep learning neural networks for a “sequence to sequence” transformation of data to solve such classical tasks, as Part-of-speech tagging (PoS), named entity recognition (NER), chunking and others. But so far accuracies of these tasks are higher for methods based on vocabularies and traditional machine learning algorithms: CRF, HMM, SVM [Gareev R., Tkachenko M., Solovyev V. et al]. These methods are based on a consistent representation of each word from a sentence as a set of binary encoded categorical features. The feature set of a word includes the word form ID from dictionary, IDs of its neighbors in the window, and a set of additional features of these words, such as: the first several characters and the last several characters of the word, the presence of capital letters, etc.

We develop a method based on deep learning neural networks for the following morphological analysis tasks:

1. PoS tagging,
2. features tagging—lexical and grammatical properties determination (except PoS).

Our method is based on a two-level representation of a sentence by individual characters level (see Section 2.1.1) and level of words (Section 2.1.2), inspired by works [Nogueira dos Santos C., Zadrozny B.], [Zhiheng H., Wei X., Kai Y.], [Plank B., Søgaard A., Goldberg Y.].

An information about words lengths, prefixes, terminations is important for some tasks such as PoS and total morphological tagging. It allows to use the additional word characters information more efficiently.

As the dataset we used the SynTagRus corpora in the original format of morphological features and its representations in the forms of Universal dependencies v1.3 and v1.4 (Section 3.2). Section 3 describes the results of comparisons of the proposed approach with other methods. At the MorphoRuEval competition the described method gained the third place under the name Sagteam on the scoreboard. In Section 4 we discuss the results obtained, as well as directions for further research on the development of the proposed method.

2. Materials and methods

Further we use the following terminology. The set of morphological categories includes part of speech (PoS), gender, number, case, and others. Each morphological category includes a set of features, for example in case of PoS category these are noun, verb, adjective, etc. A full tag is the unambiguous set of morphological features of appropriate categories for a word.

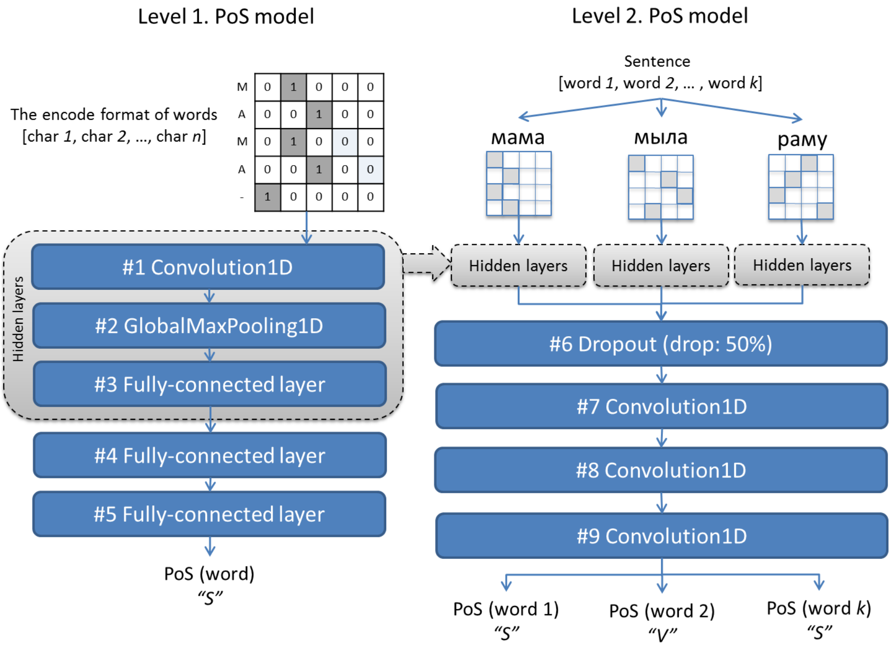
2.1. Two-level deep learning neural network model

We use two different models for the full morphological tagging: the first model for PoS-tagging and the second to predict the rest of morphological features (features tagging). These models have similar topologies and training methods. In frame of PoS-tagging task each part of speech is a separate class, classes are encoded in the one-hot manner. In frame of the features tagging task, output classes consist of all the unique combinations of lexical and grammatical properties (except PoS) that exist in the train set, one-hot encoded. Such an approach allows to decrease the computational complexity of the model. However, there might emerge combinations not presented in the training set, and such examples could be classified incorrectly.

The learning of the PoS model starts from training the first level (level 1 on figure 1) using character representation of every word of the train dataset. After that, the second level (level 2 on figure 2) is trained sentence by sentence. During the level 2 training, the first level weights are additionally tuned.

The training of features model (figure 2) is performed in a similar way, exceptthatthe input data includes PoS labels of every word predicted by the PoS-tagging model. The probabilities of PoS labels from PoS model are concatenated with hidden vector of level 1 (the Word 1 PoS, Word 2 PoS, Word *k* PoS on figure 2). The above sequential training scheme allows to re-use the symbol encoder ("hidden layers" on figures 1, 2) for other models. We implement the proposed model in Python language with the help of the Keras framework [Keras library].

Below is a detailed look at each level of the proposed topology.



M	0	1	0	0	0
A	0	0	1	0	0
M	0	1	0	0	0
A	0	0	1	0	0
-	1	0	0	0	0

Fig. 1. The model for PoS tagging

2.1.1. The first level of the model—the individual characters representation

Words, represented as sequences of one-hot encoded characters, are the input samples of the first level. We use fixed word length, short words are extended with special "null" labels from the beginning of word. The dimensionality of an input sample is L*T, where L is the maximum word length in the training set and T is the number of unique characters in the training set + 2 ("out of vocabulary" label and "null" label). The desired class for each word is a PoS label for PoS-tagging task and a features label for features-tagging task. After training the 1st level of the network gives the vector of probabilities for desired classes. The training set consists of all words as they are in the corpus, not only unique samples.

Configuration of layers on the 1st level is identical for PoS ("Level 1. PoS model" on fig. 1) and full tag models ("Level 1. Features model" on fig. 2):

- #1 Convolution1D—convolution layer, its window goes through the word characters. Window size equals to 5 without padding on the borders of the input matrix, neuron number is 1024, activation function is ReLU [Memisevic, R., & Krueger, D.];
- #2 GlobalMaxPooling—MaxPooling over the whole word;
- #3 and #4 the fully-connected layers contain 256 neurons with activation function ReLU. The #4 layer activation values are used in level 2, described further;
- #5 Fully-connected layer, size of which is equal to the number of PoS-classes and the activation function is softmax.

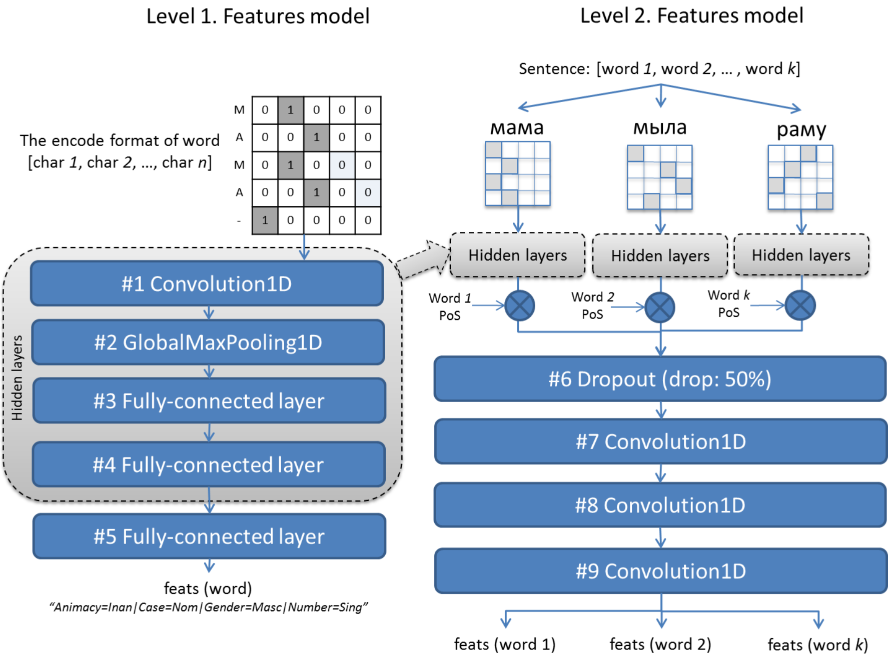


Fig. 2. The model for features tagging

2.1.2. The second level of the model—whole sentence representation

The second level input data is a whole sentence, each word of which is encoded by the activity values of a certain layer of level 1 (#3 in case of PoS-tagging and #4 in case of features tagging) in response to that word. We use fixed sentence length—the maximum length in the training set. Short sentences are extended from the beginning with "null words" consisting of "null" label characters. Such "null words" belong to special null class. The level 2 predicts labels for all words of a sentence at once.

Configuration of layers on the 2nd level (identical for PoS and full tag models):

- Layers #7 and #8 are Convolution layers, with window going through the words of the sentence. Window size equals 3 with padding on the borders of the input matrix, neurons number is 256, activation function is ReLU. A zero vector is added to the end and to the beginning of the sentence ("same" border mode in Keras).
- #9 Model output is a convolution layer, its window goes through the words, window size is 3,the neuron number is the PoS-classes number + 1 (for the zero padding),the activation function is softmax.

2.1.3. Learning configuration of two-level deep learning neural network model

We set the maximum of 300 epochs for training with early stopping: if the mean square error (MSE) stays the same or rises on a validation dataset during several consecutive epochs (15 on level 1 and 10 on level 2), training stops and neural network weights are set to the state with the minimum validation loss during training in case of PoS-tagging task, or remain at the state of the last epoch in case of the features task.

The MSE loss function is calculated for each word in the dataset on the first level training and for each padded sentence on the second level training. The optimizer is Adamax [Kingma, D., & Ba, J.]. Batch normalization function is used on the first level for activity normalization between GlobalMaxPooling and #3 layers, as well as between #4 and #5 layers. Batch size was 1024 on the 1st level and 32 on the 2nd level.

2.2. Other models for comparison

The set of well-known models were compared with the approach proposed in this paper: SVM, its extended version using Yandex.Mystem, Syntaxnet (PoS-tagging part).

2.2.1. SVM-based Approach

In this case a word is represented as the vector of word forms indices, which includes the indices of: *n* words to the left, the base word, *k* words to the right. These indices are defined on base of the learning sample dictionary. There are two rules: if the word is not in the dictionary, the ID of this word equals to 1; if in some places of window there are no words, the indices of 0 values fill these places. The ensemble of linear SVM was used, learned on base of one-vs-all strategy. The number of these classifiers equals to the number of morphological features to be defined.

2.2.2. Extension of the SVM approach

The main characteristic of this approach [Rybka, R., Sboev, A., Moloshnikov, I., Gudovskikh, D.] is to add the results of preliminary MYSTEM tagging to feature vector for the final parsing. For this purpose the MYSTEM results are transformed to tagging format of SynTagRus by the specially created converter. The list of features contains:

- All word forms from the window W;
- Tags for words of W that have been analyzed on previous steps;
- Classes of ambiguities for all words from W (+ their bigrams and trigrams). Class of ambiguity is the set of all possible tags for a word. We represent it as a concatenated tags string. For example, in case of Russian equivalent of the word “These” the sentence-example class ambiguity looks like this:

adjective|nominative_case|plural_adjective|accusative_case|plural|inanimate;

- Possible full tags for each word;
- Determined morphological features for parsed words of W;
- Possible morphological features for each unparsed word from W.

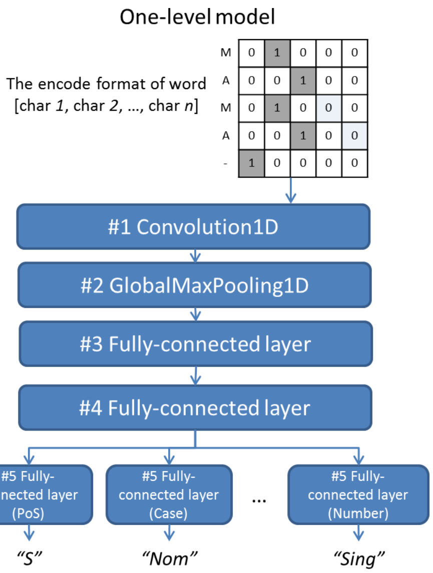
The dimension of window W equals to 7, W includes 3 words from left side and 3 words from right from the analyzed words. Words are sequentially processed from right to left. The ensemble of linear SVM was used to predict individual morphological features. Thus each classifier solves a binary classification task.

The following function is used for resulting class choice:

```
function (X, DV, M):
   # Here X={$x_1$,...,$x_K$} is the set of all possible full tags,
   # and $DV = \{dv_1,..dv_M\}$, where $dv \in [-1,+1]$, is the decision value of the classifier m
   # of the SVM ensemble
```

```
  estimation_list = [] # will contain a probability of each $x_k$ for $1 \le k \le K$
    for $k$ in range(0, K): # for each possible full tag $x_k$, now called x
      x = X[k]
      # a full tag $x_k$ is a vector $(v_1, ..., v_M)$ of morphological features $v_m \in \{0,1\}$,
      # where M is the number of morphological features in the corpus
      similar_score = 0
      different_score = 0
      for m in range(0, M): # for each morphological feature $v_m \equiv x[m]$
        if (x[m] == 1) and (DV[m] > 0): similar_score+ = DV[m]
        else: different_score = different_score + DV[m]
      estimation_score_m = similar_score – different_score
      estimation_list.append(estimation_score_m)
max_index = argmax(estimation_list) # getting the index of the highest element of estimation_list
return X[max_index] # the resulting tag is the one with the highest probability
```

2.2.3. One-level deep learning model



M	0	1	0	0	0
A	0	0	1	0	0
M	0	1	0	0	0
A	0	0	1	0	0
-	1	0	0	0	0

One-level model
The encode format of word
[char *1*, char *2*, ..., char *n*]
#1 Convolution1D
#2 GlobalMaxPooling1D
#3 Fully-connected layer
#4 Fully-connected layer
#5 Fully-connected layer (PoS)
#5 Fully-connected layer (Case)
...
#5 Fully-connected layer (Number)
"S"
"Nom"
"Sing"

Fig. 3. The model for full morphological tag for one word

Also we added another neural network model for comparison (Fig. 3), called One-level model in Table 3. At the input this model gets character word representation without its neighborhood in the sentence. The last layer consists of several parallel equal layers, each of which corresponds to a morphological category like PoS, Case, Number etc. Layer #1 is the convolution layer, performing a character-by-character pass window size of 5. The layer contains 1,024 neurons with ReLU activation function. Keras border mode is "valid". GlobalMaxPooling #2 is max-pooling over time; #3 and #4 are fully-connected layers, each having 256 ReLU neurons; #5 layers are fully connected layers with dimensionality equal to the number of features in each category (PoS, Case, Number etc.), with linear or softmax activation functions.

3. Evaluation

3.1. Prediction scores

Accuracy metric and weighted F1-score were chosen as comparison criteria. We present scores separately for testing datasets and for words not existing in the training dataset, further called out-of-vocabulary (OOV).

3.2. Used corpora

We used SynTagRus dataset with different formats of morphological features: original format contained in National Russian Corpus, Universal dependencies (UD) 1.3 and 1.4. In case of UD dataset format we used the original predefined splitting into training, testing, and developing sets. SynTagRus with original format was split into 3 parts manually. The datasets differ in number of sentences (Table 1), type of PoS-tags, and other morphological features (Table 2).

Table 1. Number of sentences and tokens in various datasets formats

SynTagRys datasets type	Number of sentences in			Number of tokens in		
	Train set	Test set	Dev set	Train set	Test set	Dev set
Original	47,980	5,923	5,331	695,255	86,163	77,249
UD-1.3	46,750	6,130	6,250	815,485	107,737	109,422
UD-1.4	48,171	6,130	6,250	850,689	108,100	109,694

Table 2: Number of unique PoS-tags, morphological features, and different full tags

SynTagRys datasets type	Number of PoS features	Number of morphological features	Number of uniques full tags	
			All corpus	Train set
Original	11	45	450	447
UD-1.3	15	36	436	433
UD-1.4	16	36	585	581

3.3. Experiments

Accuracy and F1-score metrics were calculated for the following models:

- Linear SVC (described in Section 2.2.1),
- the approach based on an extended linear SVC combined with Yandex.MyStem (described in Section 2.2.2), the window size equals 8,

- the proposed two-level deep learning neural network model approach (described in Section 2.1), with dropout and without dropout,
- the model involving a single level for full tag based only on character representation of a word (described in Section 2.2.3), called "one-level model" in Table 3. The results are presented in table 3.

Table 3. Accuracy and F1-score of different models on SynTagRus datasets in different formats

Type of SyntagRus format	Model name	all				OOV			
		POS		full		POS		full	
		accuracy	F1 score	accuracy	F1 score	accuracy	F1 score	accuracy	F1 score
Original	Extended LinearSVC: window size=±3	94.10	94.05	83.90	91.24	63.22	61.90	29.70	45.80
	Extended LinearSVC: window size=±2	94.39	94.34	85.04	91.91	62.11	60.00	30.20	46.40
	Extended LinearSVC: window size=±1	95.02	94.98	85.74	92.32	63.33	61.40	29.70	45.80
	Extended LinearSVC+ Mystem (window size of 8)	95.61	96.00	81.65	89.90	95.91	96.30	79.60	88.70
	One-level model	96.63	96.64	85.58	92.23	94.72	94.74	74.76	85.56
	Proposed approach	98.24	98.23	94.12	96.97	95.14	95.20	84.40	91.60
	Proposed approach + Dropout	**98.34**	**98.33**	**94.83**	**97.35**	**95.24**	**95.25**	**85.07**	**91.93**
Universal Dependencies 1.3	Extended LinearSVC: window size=±3	94.87	94.84	82.30	90.29	69.22	67.90	13.32	23.51
	Extended LinearSVC: window size=±2	95.20	95.17	83.33	90.91	69.17	67.47	12.60	22.38
	Extended LinearSVC: window size=±1	95.46	95.41	84.04	91.33	68.85	65.85	11.91	21.28
	One-level model	96.85	96.82	85.56	92.22	94.13	94.19	59.86	74.89
	Proposed approach	98.44	98.44	93.34	96.55	95.16	95.20	71.30	83.25
	Proposed approach + Dropout	98.49	98.49	94.31	97.07	95.07	95.09	74.48	85.37
	GOOGLE	98.27	98.27	94.01	96.92	94.21	94.35	74.12	85.13
Universal Dependencies 1.4	Extended LinearSVC: window size=±3	93.98	93.91	81.59	89.86	61.08	60.12	11.73	21.00
	Extended LinearSVC: window size=±2	94.31	94.25	82.79	90.59	60.97	59.90	12.05	21.50
	Extended LinearSVC: window size=±1	94.46	94.38	83.46	90.98	60.54	59.00	10.46	18.90
	One-level model	95.60	95.54	84.50	91.60	85.71	85.47	56.63	72.31
	Proposed approach	97.51	97.49	92.79	96.26	88.63	88.53	69.32	81.89
	Proposed approach + Dropout	**97.60**	**97.58**	**93.44**	**96.61**	88.34	88.06	**70.22**	**82.50**

Table 3 shows the following:

1) LinearSVC window size increasing does not give better accuracy.
2) The approach based on Extended LinearSVC and MyStem gives better accuracy than LinearSVC in case of out-of-vocabulary words prediction.
3) Neural network model with the one-level topology ("One-level model") gives accuracy similar to LinearSVC ones, but shows worse results in out-of-vocabulary words parsing.
4) The proposed approach shows accuracy a bit higher than the Google parser.

3.4. MorphoRuEval on Dialog 2017

As part of the competition, we used a modified version of the two-level model. We add the Batch normalization layers between layers #2 and #3 and between layers #4 and #5 in level 1. The model is trained on the GICRYA corpus, provided by the organizers. The corpus was divided into a training (90%) and validation set (10%). Testing was performed on three datasets (not disclosed to the competition participants): news, posts in a social network ("social media" in Table 4) and fiction literature. For each dataset two tasks were graded, full tagging and lemmatization, and two accuracy measures were evaluated for each task, the ratio of words correctly classified and the ratio of sentences completely correct.

Table 4. The results of the model for each measure compared to a few leaders

Dataset	Task	Accuracy measure	First place (%)	Second place (%)	This study (%)	Fourth place (%)	Fifth place (%)
News	Full tagging	Accuracy on words	93,71	93,99	93,35	93,83	90,52
		Accuracy on sentences	64,80	63,13	55,03	61,45	44,41
	Lemmatization	Accuracy on word forms		92,96	81,6	93,01	
		Accuracy on sentences		56,42	17,04	54,19	
Social Media	Full tagging	words	92,29	92,39	92,42	91,49	89,55
		sentences	65,85	64,08	63,56	61,44	51,41
	Lemmatization	word forms		91,69	82,8	90,97	
		sentences		61,09	35,92	60,21	
Fiction literature	Full tagging	words	94,16	92,87	92,16	92,4	90,13
		sentences	65,23	60,91	56,6	60,15	48,48
	Lemmatization	word forms		92,01	77,78	91,46	
		sentences		57,11	22,08	55,08	
Mean	Full tagging	words	93,39	93,08	92,64	92,57	90,07
		sentences	65,29	62,71	58,4	61,01	48,1
	Lemmatization	word forms		92,22	80,73	91,81	
		sentences		58,21	25,01	56,49	

4. Conclusion

The presented results demonstrate the great potential of complicated deep learning models compared to traditional SVM ones. The approach on base of MYSTEM is more effective in case of words not presented in the training set. This fact is expected since the latter approach is based on common dictionaries and linguistic rules without tuning to any corpus. As a result it loses in comparison to deep learning models in specific cases. For practical needs it would be useful to unite these approaches in a common morphological parser to increase the universality and the accuracy of parsing.

Acknowledgments

The reported study was funded by RFBR according to the research project № 16-37-00214

References

1. *Gareev R., Tkachenko M., Solovyev V., Simanovsky A., Ivanov V.* (2013), Introducing baselines for Russian named entity recognition. Volume 7816 of the series Lecture Notes in Computer Science, pp. 329–342.
2. Keras library [Online]. Available: http://keras.io
3. *Kingma, D., & Ba, J.* (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
4. *Memisevic, R., & Krueger, D.* (2014). Zero-bias autoencoders and the benefits of co-adapting features. stat, 1050, 13.
5. *Nogueira dos Santos C., Zadrozny B.* (2014), Learning Character-level Representations for Part-of-Speech Tagging, Proceedings of the 31st International Conference on Machine Learning (ICML) — Volume 32, Beijing, China, pp. II-1818 — II-1826.
6. *Plank B., Søgaard A., Goldberg Y.* (2016), Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 412–418.
7. Russian national corpus (2016) [Online]. Available: http://ruscorpora.ru/en
8. *Rybka, R., Sboev, A., Moloshnikov, I., Gudovskikh, D.* (2015, November). Morphosyntactic parsing based on neural networks and corpus data. In Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015 (pp. 89–95). IEEE.
9. *Zhiheng H., Wei X., Kai Y.* (2015), Bidirectional LSTM-CRF Models for Sequence Tagging, availiable at: arXiv:1508.01991.

SEMANTIC ROLE LABELING WITH NEURAL NETWORKS FOR TEXTS IN RUSSIAN

Shelmanov A. O. (shelmanov@isa.ru),
Devyatkin D. A. (devyatkin@isa.ru)

Federal Research Center "Computer Science and Control"
of Russian Academy of Sciences, Moscow, Russia

We present and evaluate neural network models for semantic role labeling of texts in Russian. The benchmark for evaluation and training was prepared on the basis of the FrameBank corpus. The paper addresses different aspects of learning a neural network model for semantic role labeling on different feature sets including syntactic features acquired with the help of SyntaxNet. In this work, we rely on architecture engineering and atomic features instead of commonly used feature engineering. We investigate the ability of learning a model for labeling arguments of "unknown" predicates that are not present in a training set using word embeddings as features for the replacement of predicate lemmas. We publish the prepared benchmark and the models. The experimental results can be used as a baseline for further research in semantic role labeling of texts in Russian.

Keywords: semantic parsing, semantic role labeling, frame parsing, neural network, word embeddings, deep learning

1. Introduction

Semantic role labeling (SRL) is a useful type of linguistic analysis that maps varying low-level syntactic representations of sentences to more abstract argument-predicate structures. Predicates in these structures are words that express situations, they are verbs, verbal nouns, and verb forms. Arguments are words and phrases (often noun phrases) that play a role in a situation expressed by a predicate. These semantic roles capture meaning of arguments and explicitly present meaningful aspects encoded in the sentence by an author. The significance of semantic role annotation lies in the fact that such abstract semantic representations naturally can be applied for a variety of natural language processing tasks, which require comparison of texts by their meaning: question answering [Shen and Lapata, 2007], information extraction [Christensen et al., 2011], information search [Osipov et al., 2014], machine translation [Liu and Gildea, 2010], and others.

The majority of the state-of-the-art methods for SRL rely on supervised learning techniques that require a lot of annotated data. This is a problem for developing a good SRL system, since creating such an annotated resource is a very expensive and difficult task. Such resources have been created for several languages. For today, the most used and researched resources are FrameNet [Baker et al., 1998] and Propbank [Kingsbury and Palmer, 2002]—corpora that provide SRL annotations for English texts. For long time, there was no such a resource for Russian. Although several semantic parsers that produce SRL-like annotations were presented in the past, they mostly relied on hand-crafted rules and dictionaries [Sokirko, 2001], as well as on training on automatically annotated corpus [Shelmanov and Smirnov, 2014]. However, the recent release of FrameBank corpus [Lyashevskaya, 2012; Lyashevskaya and Kashkin, 2015] enables new capabilities of using machine learning techniques for creating semantic role labelers that work with Russian language and for new fundamental research in this direction. In this work, we investigate the ability of training a semantic role labeler based on neural networks using various types of linguistic features and word embeddings [Le and Mikolov, 2014].

The FrameBank provides the hierarchical role schema, the lexicon with predicates that mostly are verbs (and verb forms), and the partially annotated text corpus for more than 800 predicates. We note that the verb coverage by examples of the corpus is still not very high. This encourages us to develop semi-supervised approach to improving the parser capabilities of annotating sentences with "unknown" predicates that are not present in the training set. Therefore, in addition of creating and evaluating neural network models for SRL we also investigate the ability of using word embeddings to mitigate the problem of low verb coverage.

The main contributions of this paper are the following:

1. The openly available benchmark for evaluation of semantic parsers for Russian language based on FrameBank corpus[1].
2. The openly-available neural network models for semantic role labeling trained on FrameBank and evaluated on different feature sets.
3. The method for processing "unknown" predicates based on word embeddings.

2. Related Work

One of the first methods for SRL presented in [Gildea and Jurafsky, 2002] was based on a simple statistical model. Since then, more sophisticated machine learning techniques have been elaborated very quickly. Several shared tasks CoNLL-2004, 2005, 2008, and 2009 [Hajic et al., 2009] set up some common benchmarks and revealed useful machine learning approaches, in which authors investigated different features sets, task decomposition methods, and global inference techniques. Early works devoted to SRL heavily relied on complex feature engineering. The advances in neural network training as well as in learning of meaningful representations of words sparked new interest to problem of SRL. In many recent works, researchers propose new neural network approaches based on architecture engineering. It was revealed that neural networks do not need complex features, instead they can rely on atomic features or even on very low-level representations like tokens or n-grams. Such models often significantly outperform the traditional ones. In the rest of the section, we review the recent works devoted to SRL for English and Russian.

One of the first well-known publications, in which feature engineering was replaced by an architecture engineering, is [Collobert et al., 2011]. The researchers presented and applied a single neural network model to various natural language

[1] http://nlp.isa.ru/framebank_parser

processing tasks including part-of-speech tagging, named entity recognition, and semantic role labeling. They showed that this approach allows to reduce domain and task specific feature engineering. The main idea of this work lies in exploiting latent interactions between features in big and mostly unlabeled training sets.

The paper [Roth and Lapata, 2016] proposes a novel model for SRL based on recurrent neural network. The researchers claim that complex syntactic structures are not analyzed well by baseline approaches. They proposed a model that processes subsequences of lexicalized dependency paths and learns suitable embedding representations of them. The researchers empirically showed that such embeddings can improve results over the previous baseline SRL approaches.

In the similar way, [FitzGerald et al., 2015] presented a new model for SRL, in which arguments and semantic roles are jointly embedded in a shared vector space for a given predicate. This model utilizes finer-grained semantic similarity between roles. The researchers trained a neural network to approximate the potential functions of a graphical model designed for the SRL task and used this network to build embeddings. They showed that the proposed model can learn jointly from PropBank and FrameNet to achieve performance improvements on the smaller FrameNet dataset.

In [Foland and Martin, 2015], authors proposed a method for SRL based on convolutional and time-domain neural networks. The method takes into account features derived from a dependency parser output. The authors explored the benefits of adding increasingly more complex dependency-based features to the model. The proposed method demonstrated state-of-the-art performance and low computational requirements.

Recently, several works proposed end-to-end SRL approaches that do not require syntactic features. These approaches allow to avoid losing information between different stages of text processing.

In [Marcheggiani et al., 2017], researchers proposed a simple syntax-agnostic model for dependency-based SRL. That model predicts predicate-argument dependencies relying on states of a bidirectional LSTM encoder [Hochreiter and Schmidhuber, 1997]. The authors showed that sufficient accuracy on English texts can be achieved even without syntactic information using only local inference. It was also approved that the model is more robust on the standard out-of-domain test set than the baselines.

Similar approach was proposed in [Zhou and Xu, 2015]. The researchers applied a model based on bidirectional recurrent network for end-to-end SRL. They did not use any syntactic information but relied only on original text as the input features. The model was evaluated on SRL task of CoNLL-2005 and coreference resolution task of CoNLL-2012. It outperformed the previous state-of-the-art ensemble models. The authors revealed that the proposed model is better at processing longer sentences than the baseline approaches.

It is also worth noting great interest to joint modeling of syntax and semantics in many works devoted to SRL. For example, in [Swayamdipta et al., 2016], a transition-based model for SRL that jointly produces syntactic and semantic dependencies was presented. The model is based on a stack of LSTM cells and is used for representation of the entire algorithm state. The researchers also proposed a greedy inference algorithm, which works in linear time. They obtained the best published parsing performance among models that jointly learn syntax and semantics on the CoNLL-2008, 2009 datasets.

There are a few works devoted to semantic role labeling of Russian language texts. In the previous work, we used rule and dictionary-based semantic parser for creating automatically annotated corpus for training a model for SRL [Shelmanov and Smirnov, 2014]. In [Kuznetsov, 2015; Kuznetsov, 2016], SVM-based semantic role labeler was trained on FrameBank corpus. The corpus was supplemented by syntactic features generated by the pipeline presented in [Sharoff and Nivre, 2011]. The author also performed clustering of lexis features to extract additional semantic information from the corpus and used ILP-optimization approach for post processing. This work is based on the pre-release version of the FrameBank corpus and does not provide the tools for the data preparation, modeling, and evaluation. In this work, author did not use neural networks and word embeddings as features mostly relying on feature engineering.

In our work, instead of feature engineering, we use atomic features with word embeddings and neural networks. We also research the problem of semantic role labeling for "unknown" predicates (out-of-domain predicates) and propose the simple approach to that problem. We publish the benchmark for model construction and evaluation on the FrameBank corpus.

3. Neural Network Models for FrameBank Parsing

We present two neural network models for semantic role labeling. These models mostly diverge in the way different feature types are aggregated. We used the following features:

Categorical:

1) Various types of morphological features of both an argument and a predicate: part of speech, grammar case, animacy, verb form, time, passiveness, and others. ("morph").
2) Relative position of an argument in a sentence with respect to a predicate. ("rel_pos").
3) Predicate lemma ("pred_lemma").
4) Preposition of an argument extracted from a syntax tree ("arg_prep").
5) Name of a syntax link from argument to its parent in a syntax tree ("synt_link").

Embeddings:

1) Embedding of an argument lemma ("arg_emeddings").
2) Embedding of a predicate lemma ("pred_emeddings").

The first neural network model has the simple architecture that acquires all features of an argument: sparse and dense, as a single vector and propagates them through three dense layers. The two hidden layers have ReLU activations and the output layer has softmax activation. The softmax activation is a standard way of producing final probabilities of classes in a multinomial classification task. The ReLU activation is a rectifier function that propagates only positive signal through a network. This activation function is convenient since it simplifies training of deep architectures and results in lesser overfitting effect than many other functions. In the hidden layers, we use batch normalization [Ioffe and Szegedy, 2015]. In this technique, inputs of layers are normalized in each mini-batch, which drastically increases the training speed of networks and also regularizes them. The network also has two dropout layers for additional regularization. We will refer to this model as "simple".

The second neural network is intended to handle embeddings and categorical features more intelligently than the "simple" one. The problem of processing the both types of features lies in their different nature. The categorical features are sparse, therefore, merging them with embeddings within one dense layer would result in a big number of parameters. The better way of handling this case is to embed sparse categorical features first and merge them later. Therefore, the complex model has the same types of layers but the first layer is split into several chunks: a chunk for categorical features, a chunk for an argument embedding (if it is present in a feature set), and a chunk for a predicate embedding (if it is present in a feature set). Such an architecture is much smaller than the "simple" one in terms of parameters, thus, it overfits less and is trained faster. We will refer to this model as "complex". The Figure 1 depicts the neural network architectures.

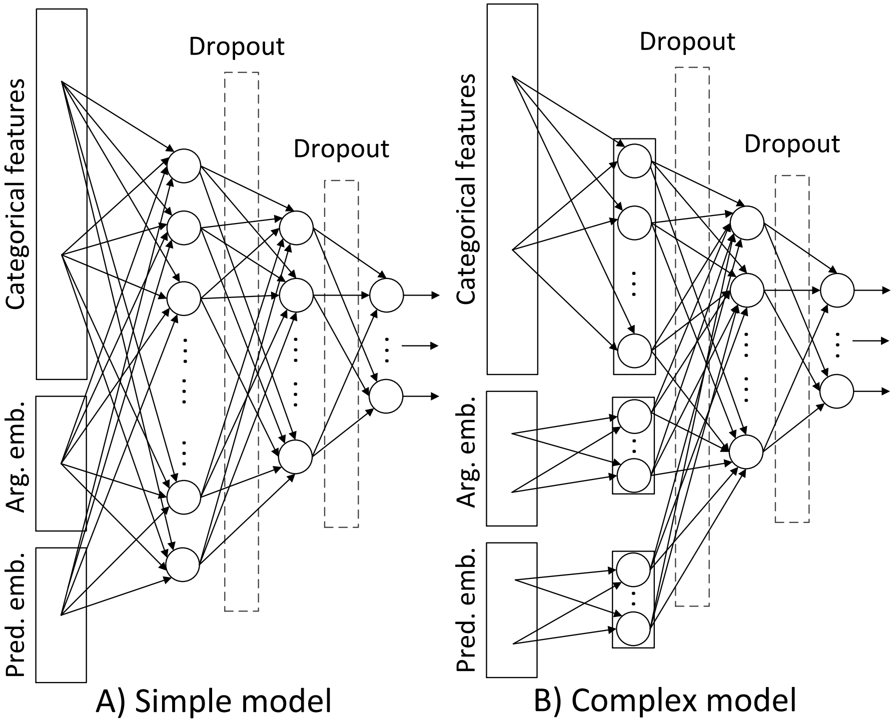


Figure 1. Architectures of neural network models

We compile these models with Adam optimizer [Kingma and Ba, 2015] and a standard categorical cross entropy loss function. These models and different subsets of the aforementioned features are used for labeling of arguments of "known" predicates.

For labeling arguments of "unknown" predicates, we also use the similar architectures. However, in this setting we cannot rely on predicate lemma feature, since there will be no lemma in the test set known by the model. In this setting, predicate embeddings should give the most significant impact on a network performance. Embeddings, due to the way they are built, encode semantic similarities of words in a low dimensional vector space [Le and Mikolov, 2014]. Many text processing methods that have been recently developed rely heavily on this remarkable property of embeddings and demonstrate its great usefulness. We investigate the ability of substituting predicate lemma feature in SRL parser by its embedding. Embeddings are built in an unsupervised manner on a huge unlabeled corpus, so model does not need to see every predicate lemma in a small semantically labeled training set to obtain its embedding. Since such embeddings encode similarities between words, they could also encode similarities between frame structures of predicates. Therefore, we can use training examples of "known" predicates to infer the frame structure of "unknown" predicates that are similar to the former in an embedding vector space. The bigger the similarity, the more precise we can restore frame structures of "unknown" predicates.

In the setting for "unknown" predicates, we additionally used early stopping in the training procedure since it becomes useless to tune fixed number of epochs for out-of-domain test set.

4. Experiments

4.1. Experiment Setup

We used the publicly released version of FrameBank corpus[2]. The corpus contains annotated text examples that consist of multiple sentences. Tokens in the sentences are annotated with morphological and some other features. The role and the predicate annotations are separated from the texts. The original version of the corpus does not contain explicit exact mapping between role annotations and tokens or text spans. To mitigate this problem, we created the automatic tool for mapping predicates and arguments with core roles to text tokens.

To create the syntax annotation for FrameBank, we used Google's SyntaxNet parser[3] [Andor et al., 2016]. This parser was trained on SyntagRus treebank [Nivre et al., 2008] and provides high quality parsing for Russian texts according to [Alberti et al., 2017]. We used dockerized version of SyntaxNet with a model for Russian[4,5]. The parser creates a fully connected dependency tree for a sentence with syntax tags on every parent-child link. The syntax structure corresponds to well-known Universal dependencies format [Nivre et al., 2016].

After the mapping procedure, we obtained the corpus that contains examples for 803 predicates. We selected the subcorpus by keeping only predicates that have at least 10 examples. This results in 572 predicates left in the subcorpus. We also filtered out

[2] https://github.com/olesar/framebank

[3] https://github.com/tensorflow/models/tree/master/syntaxnet

[4] https://github.com/IINemo/docker-syntaxnet_rus

[5] https://hub.docker.com/r/inemo/syntaxnet_rus

arguments with infrequent semantic roles and preprocessed erroneous role labels that do not correspond to the role ontology of FrameBank published in [Kashkin and Lyashevskaya, 2013]. The final version of the whole experimental dataset contains 53,151 examples with 44 different semantic roles.

The word embeddings used in our experiments are provided by RusVectores 2.0[6] [Kutuzov and Andreev, 2015]. They were pre-trained on Russian national corpus and have 300 dimensions. We note high quality of the model; however, we also note that a large portion of predicates (verbs) presented in FrameBank are not covered by it. Therefore, more than 17,000 examples in our dataset have zero predicate embeddings.

The hyperparameters of the proposed neural network models on different features sets were tuned using the greedy strategy. We mostly tuned dropout ratio, the size of internal dense layers, and a number of training epochs.

For the simple baseline, we use a parser that assigns the most frequent semantic role to every argument in the test set. Obviously, this baseline has low performance, but it shows the skewness in the evaluation set, which reflects the complexity of the task and the impact of other models.

We evaluated our models using macro and micro F_1 score. We note that our results are not directly comparable with the results presented in [Kuznetsov, 2015]. This is due to the fact that the author used different annotation scheme and different prerelease version of FrameBank corpus with unknown preprocessing procedures.

4.2. Evaluating Models on "Known" Predicates

In the first experiment, we evaluate our models on different feature sets: lexis, morphological, syntactic, and word embeddings. In each feature set we also use relative position feature. The performance of the models is assessed using five-fold cross validation on the selected subcorpus of FrameBank. The evaluation results are presented in Table 1.

Table 1. Performance of the models on different feature sets

Model + feature set	**Macro F_1-score, %**	**Micro F_1-score, %**
Baseline	0.5 ± 0.0	11.6 ± 0.2
Simple + morph	22.8 ± 0.6	35.4 ± 0.3
Simple + morph + pred_lemma	71.2 ± 0.6	76.1 ± 0.5
Simple + morph + pred_emeddings	62.0 ± 0.4	65.2 ± 0.3
Simple + morph + pred_lemma + arg_prep	75.9 ± 0.4	79.2 ± 0.2
Simple + morph + pred_lemma + arg_prep + synt_link	76.8 ± 0.5	80.3 ± 0.3
Simple + morph + pred_lemma + arg_prep + synt_link + arg_embeddings + pred_embeddings	78.6 ± 0.4	81.8 ± 0.2

[6] http://rusvectores.org/en

Model + feature set	**Macro F_1-score, %**	**Micro F_1-score, %**
Complex + morph + synt + pred_lemma + arg_embeddings + pred_embeddings	**79.2 ± 0.3**	**82.3 ± 0.2**

Since the evaluation dataset is not very unbalanced, the baseline that marks the dominant class has a very low performance. Adding morphological features of predicates and arguments results in a substantial improvment over the baseline: Δmicro $F_1 = 23.8\%$. This setting shows the importance of low-level linguistic features in semantic role labeling without appealing to any semantics of arguments and predicates. This performance could be achieved without any knowledge about meaning of predicates or arguments and syntactic information. With adding predicate lemmas, we drastically improve performance of labeling by Δmicro F_1=40.7%, which is not surprising. Since frame structures are invoked by a predicate that represents a situation, roles can be very specific to predicates. Without knowledge of which predicate invoked the current frame, in many cases, it is impossible to distinguish roles of morphologically similar arguments. The results of the setting, in which we substitute predicate lemma with its embeddings, show that the performance drop without predicate lemmas is not very big, when at least embeddings of predicates are present. This enables the ability of building a model for "unknown" predicates relying on properties of word embeddings.

In the next setting, the feature set is composed of morphological features, predicate lemmas, and argument preposition. The preposition in Russian is considered to be very important for semantic role labeling. We observe an Δmicro $F_1 = 3.1\%$ increase compared to the model that does not take it into account, which is very significant for building a good semantic parser. Adding names of parent syntax links of arguments as features extends this improvement by another percent. We used only basic syntactic features: preposition and the parent link, whereas it is also worth adding, e.g., the syntactic path from argument to predicate as suggested in many previous works. We leave this for the future work, since it would require comparison of many different embedding techniques for a very sparse space of syntactic paths. We also note that although the syntactic features are important for building a good SRL model, they do not drastically increase the performance of the parser. Following several techniques presented in related work that suggest syntax agnostic models for English, we consider the task of creation an accurate model for Russian without appealing to syntactic parsing also feasible.

Adding embeddings of arguments and predicates to the rest of the features yields the best results. The "simple" model as expected gives the smallest performance gain Δmicro $F_1 = 1.5\%$. Adding embeddings directly as additional dimensions results in a big growth of a number of parameters. Therefore, such a network tends to overfitting. The "complex" model due to its architecture is twice as smaller in terms of parameters compared to the "simple" one. It gives another small but significant performance improvement Δmicro $F_1 = 0.5\%$ compared to the "simple" model on the full feature set. It also trains and runs much faster than the "simple" model.

4.3. Evaluating Models on "Unknown" Predicates

In the second experiment, we research the importance of word embeddings in the task of labeling arguments of "unknown" predicates. For this setting, we split the selected subcorpus of FrameBank in two parts: training and testing in such a way that the part for testing contains only predicates that are absent from the part for training. We perform evaluations for two different split methods. In the first split, the test part is composed from examples for predicates that have highly similar predicates in the training part. For that, cosine similarity of every two predicate embeddings is calculated. The top 27 similar pairs of predicates are distributed into different parts of corpus. In this case, we get 49,709 training and 3,442 testing examples. Such a split represents the good case, in which semantic similarity of "unknown" predicates to "known" ones can be captured by their word embeddings. This case should be easy for the models. In the second split, in a contrary, we compose the test part from predicates that are least similar to any of the "known" predicates. This split yields 50,093 training examples, and 3,058 testing examples with 21 predicates in the test set. This case should be the hardest for the models to handle. In this experiment, we do not perform cross-validation. Instead, we train models five times with different random seeds and test them on the prepared holdout. This does not prevent overfitting but alleviates the problem of randomness of model training.

We compare "simple" model with all categorical features and without embeddings, the "complex" model with categorical features and only argument embeddings, and the "complex" model with categorical features, as well as argument and predicate embeddings. The evaluation results are presented in Table 2 and 3.

Table 2. Evaluation of the models on the "unknown" predicates in the "good" split

Model + feature set	**Macro F_1-score, %**	**Micro F_1-score, %**
Baseline	0.4	9.6
Simple	13.7 ± 0.4	24.6 ± 0.3
Complex + arg_embeddings	19.4 ± 0.3	31.9 ± 0.5
Complex + arg_pred_embeddings	41.4 ± 0.7	66.7 ± 1.1

Table 3. Evaluation of the models on the "unknown" predicates in the "bad" split

Model + feature set	**Macro F_1-score, %**	**Micro F_1-score, %**
Baseline	0.7	13.2
Simple	9.1 ± 0.2	24.8 ± 0.5
Complex + arg_embeddings	14.5 ± 0.7	27.2 ± 0.1
Complex + arg_pred_embeddings	24.1 ± 1.5	41.4 ± 2.2

The results show that there is a substantial performance drop in macro and micro scores on the "unknown" predicates. However, we see that on completely unseen predicates the complex model with embeddings shows a decent micro score. The model on the good split shows expectedly better results than on the bad split. This confirms the significance of the presence in the training set of predicates that are similar to “unknown” ones in the embedding vector space. However, we note that even on a “bad” split the model with embeddings shows much better performance compared to the “simple” model that uses only morphological and syntactic categorical features. We should also note again that the substantial part of predicate embeddings in the training set are zeros due to already mentioned limitations of used language model. This definitely affects the performance of the SRL models. In the future work, it is worth training neural networks using more complete language models.

5. Conclusion and Future Work

We presented the neural network models for semantic role labeling of Russian texts. We also presented the basic benchmark based on FrameBank corpus for evaluation of parsers for SRL. Both the models and the benchmark are openly available[7]. The proposed models were evaluated on different feature sets. The achieved scores could be used as a baseline for the future research. We also investigated the method for training a labeler for arguments of “unknown” predicates using word embeddings. We demonstrate that good embeddings are essential for building a model for “unknown” predicates, however, it is not enough to approach the performance of models trained and tested on in-domain data.

In this work, we did not provide the semantic argument identifier and did not perform the global inference step in the SRL parser. The reason for that consists in the fact that FrameBank corpus provides very sparse annotations (not every argument in sentences is labeled). Therefore, learning inference procedure using straightforward approach is hardly possible. However, in the future work, we are looking forward to adapt self-learning techniques on the partially annotated data and use integer linear programming inference that does not require additional training to further boost the performance of the parser.

Acknowledgments

The project is supported by the Russian Foundation for Basic Research, project number: 16-37-00425 “mol_a”.

References

1. *Alberti C., Andor D., Bogatyy I., Collins M., Gillick D., Kong L., et al.* (2017), SyntaxNet Models for the CoNLL 2017 Shared Task. arXiv preprint arXiv:1703.04929.
2. *Andor D., Alberti C., Weiss D., Severyn A., Presta A., Ganchev K., Petrov S., and Collins M.* (2016), Globally normalized transition-based neural networks. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2442–2452.
3. *Baker C. F., Fillmore C. J., and Lowe J. B.* (1998), The Berkeley FrameNet project. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, pp. 86–90.

[7] http://nlp.isa.ru/framebank_parser

4. *Christensen J., Mausam, Soderland S., and Etzioni O.* (2011), An analysis of open information extraction based on semantic role labeling. In Proceedings of the sixth international conference on Knowledge capture, pp. 113–120.
5. *Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., and Kuksa P.* (2011), Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug):2493–2537.
6. *FitzGerald N., Täckström O., Ganchev K., and Das D.* (2015), Semantic role labeling with neural network factors. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 960–970.
7. *Foland W. and Martin J.* (2015), Dependency-based semantic role labeling using convolutional neural networks. In Proceedings of the Fourth Joint Conference on Lexical and Computa-tional Semantics, pp. 279–288.
8. *Gildea D. and Jurafsky D.* (2002), Automatic labeling of semantic roles. Computational linguistics, 28(3):245–288.
9. *Hajic J., Ciaramita M., Johansson R., Kawahara D., Mart M. A., Màrquez L., Meyers A., Nivre J., Padó S., Štepánek J., et al.* (2009), The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–18.
10. *Hochreiter S. and Schmidhuber J.* (1997), Long short-term memory. Neural computation, 9(8):1735–1780.
11. *Ioffe S. and Szegedy C.* (2015), Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pp. 448–456.
12. *Kashkin E. and Lyashevskaya O.* (2013), Semantic roles and construction net in Russian FrameBank [semanticheskie roli i set'konstrukcij v sisteme framebank]. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" (2014), volume 1, pp. 297–311, (in Russian).
13. *Kingma D. and Ba J.* (2015), Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR).
14. *Kingsbury P. and Palmer M.* (2002), From treebank to PropBank. In Proceedings of the International Conference on Language Resources and Evaluation, pp. 1989–1993.
15. *Kutuzov A. and Andreev I.* (2015), Texts in, meaning out: Neural language models in semantic similarity tasks for Russian, volume 2, pp. 133–144.
16. *Kuznetsov I.* (2015), Semantic role labeling for Russian language based on Russian FrameBank. In Proceedings of International Conference on Analysis of Images, Social Networks and Texts, pp. 333–338.
17. *Kuznetsov I.* (2016), Automatic semantic role labelling in Russian language [Avtomaticheskaya razmetka semanticheskih roley v russkom iazike], PhD thesis (in Russian).
18. *Le Q. V. and Mikolov T.* (2014), Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, volume 14, pp. 1188–1196.
19. *Liu D. and Gildea D.* (2010), Semantic role features for machine translation. In Proceedings of the 23rd International Conference on Computational Linguistics, pp. 716–724.

20. *Lyashevskaya O.* (2012), Dictionary of valencies meets corpus annotation: a case of Russian FrameBank. In Proceedings of the 15th EURALEX International Congress, volume 15.
21. *Lyashevskaya O. and Kashkin E.* (2015), FrameBank: a database of Russian lexical constructions. In International Conference on Analysis of Images, Social Networks and Texts, pp. 350–360.
22. *Marcheggiani D., Frolov A., and Titov I.* (2017), A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. arXiv preprint arXiv:1701.02593.
23. *Nivre J., Boguslavsky I. M., and Iomdin L. L.* (2008), Parsing the SynTagRus treebank of Russian. In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pp. 641–648.
24. *Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajic J., Manning C. D., McDonald R., Petrov S., Pyysalo S., Silveira N., et al.* (2016), Universal dependencies v1: A multilingual treebank collection. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), pp. 1659–1666.
25. *Osipov G., Smirnov I., Tikhomirov I., Sochenkov I., Shelmanov A., and Shvets A.* (2014), Information retrieval for R&D support. In Professional Search in the Modern World, pp. 45–69.
26. *Roth M. and Lapata M.* (2016), Neural semantic role labeling with dependency path embeddings. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1192–1202.
27. *Sharoff S. and Nivre J.* (2011), The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" 2011.
28. *Shelmanov A. O. and Smirnov I. V.* (2014), Methods for semantic role labeling of Russian texts. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" (2014), number 13, pp. 607–620.
29. *Shen D. and Lapata M.* (2007), Using semantic roles to improve question answering. In Proceedings of the Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning, pp. 12–21.
30. *Sokirko A.* (2001), A short description of Dialing Project, available at URL: http://www.aot.ru/docs/sokirko/sokirko-candid-eng.html
31. *Swayamdipta S., Ballesteros M., Dyer C., and Smith N. A.* (2016), Greedy, joint syntactic-semantic parsing with stack LSTMs. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, pp. 187–197.
32. *Zhou J. and Xu W.* (2015), End-to-end learning of semantic role labeling using recurrent neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1127–1137.

EXTRACTING CHARACTER NETWORKS TO EXPLORE LITERARY PLOT DYNAMICS

Skorinkin D. A. (dskorinkin@hse.ru)

Higher School of Economics, Moscow, Russia

In this paper we apply network analysis to the study of literature. At the first stage of our investigation we automatically extract networks (graphs) of characters for each part of Leo Tolstoy's novel *War and peace* using two different techniques for network creation. Then we evaluate these two techniques against a set of manually created gold standard networks. Finally, we use the method that demonstrated better performance in our evaluation to test a literary hypothesis about Tolstoy's novel. The hypotheses we intended to prove was that the parts of the novel describing war (i.e. those where the battlefield or military units are the primary settings), have statistically lower density of interaction between characters, resulting in lower network density, higher network diameters and lesser average node degrees. By showing this correlation we mean to demonstrate the applicability of network analysis to computational research of fictional narrative (e.g. detection of tension changes in the plot).

Key words: networks, network theory, social network analysis, literary network analysis, graph models, digital literary studies, Russian literature

1. Introduction

Over the last decades network analysis found successful applications to a great variety of fields ranging from sociology and political science to criminology and epidemiology. In recent years literary scholars, whose objects of study are also convertible to vertices and edges, turned their attention to graph[1] theory and started actively borrowing methods from social network analysis.

It has been shown that networks of fictional characters are similar to those of real social networks [Alberich et al., 2002] and share certain characteristics (e.g. power law distributions) with all other complex network types [Park, Kim, 2013]. Network theory allowed researchers to make novel observations about the composition and plot of literary pieces [Elson et al., 2010], [Moretti, 2011] and get new "insight into the roles of characters in the story" [Agarwal, Corvalan et al., 2012].

However, this ability to look at certain work of fiction from a different angle is not the only advantage of such graph-based formalization. Combined with various NLP-related techniques for automatic network extraction (some of which are implemented in this study), network analysis also opens the doors to large-scale analysis of fiction.

[1] In this paper we treat 'network' and 'graph' as synonymous words both meaning 'a set of vertices connected by edges'.

Such analysis, often referred to as ‘distant reading’ [Moretti, 2013], ‘scalable reading’ [Weitin, 2017] or ‘macroanalysis’ [Jockers, 2013], has been a point of heated debates in literary studies in recent years. The proponents of large-scale computational analysis of literature claim that close reading and precise analysis of particular pre-selected texts, traditional for literary scholars of the past, can no longer be considered sufficient for scientific research, as these approaches are only applicable to very narrow selections of works (usually the so-called *canon*, itself a very ill-defined and arguable concept). They suggest literary scholars should ‘learn how *not* to read’ the texts they study [Moretti, 2013] and ‘start counting, graphing, and mapping them instead’ [Moretti, 2007]. And although there is a fair share of criticism towards this approach, the fact remains that even a single literary movement in a single national literature usually generates more text than a single person can read, much less analyze, in his lifetime[2].

2. Related work

There has been a number of research on extraction and exploration of fictional networks. [Agarwal, Kotalwar et al., 2013] extract social events, i.e. interactions between characters or perceptions of one character by another, form Carrol’s Alice in Wonderland; [Ardanuy, Sporleder, 2015] use networks to perform genre classification of XIX century novels; [Lee, Yeung, 2012] investigate the structure of the Old Testament linking people to places thus creating spatio-personal networks; [Elson et al., 2010] explore 60 British XIX century novels through conversational networks generated from dialogues of the characters. That latter work, presented at the ACL 2010 conference, deserves a separate mention. Unlike many others, [Elson et al., 2010] do not limit themselves to network extraction and evaluation against some gold standard; their main goal is to use structural properties of networks to disprove an influential literary theory(hypothesis). The hypothesis claimed that ‘rural’ novels reflected typical social structure of a village with its close-knit community of people familiar to each other, whereas ‘urban’ novels demonstrated more complex social networks with several communities, lesser overall density and a plethora of ‘weak ties’; and that therefore the importance and amount of dialogue decreased as novels shifted from rural to urban settings after the industrial revolution. However, [Elson et al., 2010] did not find this to be the case.

In our investigation we also try to employ network parameters and network statistics as a means of testing a literary hypothesis. An additional motivation for our study was lack of literary network research made on Russian material, the only notable exception being [Bocharov, Bodrova, 2014]. That latter work, however, does not go beyond basic network extraction and evaluation, and its authors made no attempt to prove any literary theory or hypothesis.

[2] For instance, it is estimated that Victorian novels alone make up a corpus of about 60,000 texts [Moretti, 2013]

3. Hypothesis and relevant network metrics

Much like [Elson et al., 2010], we chose to study the relation between the settings in which the plot unfolds and the structural properties of the character network. However, in our case the main opposition was not 'urban' vs 'rural', but 'war' vs 'peace'. This antithesis not only gave the novel its ever-famous title[3], it is certainly among the pillars of the whole work. One of the most acclaimed Tolstoy scholars Boris Eikhenbaum spoke of *War and peace* as a novel where "The Iliad" (i.e. war) must "follow the Odissey" (i.e. peace) [Eikhenbaum, 2009 (1931), p. 497]; notable American slavist Gary Saul Morson calls this the "central opposition" of the book and claims that "the salon and the battlefield represent the extremes of order and chaos — of 'peace' and 'war' — in War and peace" [Morson, 1987, p. 97]. Note that Morson uses spatial settings — salon and battlefield — as metonymic labels for the complex concepts of 'war' and 'peace'; this indicates that spatial dynamics of narrative is the primary marker for switches between these two 'extremes'. And indeed, chapters (главы) and even entire parts (части) of Tolstoy's *War and peace* can be fairly easily subdivided into 'peaceful' and 'wartime' ones by simply looking at the space in which the plot unfolds.

This contrast between war and peace can be observed on many levels, among which the level (and intensity) of character interactions. It were changes at this level that we hoped to detect with network analysis. We had two reasons to believe that such interactions should be visibly influenced by settings:

1. Research on dramatic texts shows that tragedies tend to have lower density of networks [Trilcke et al., 2015b], and a possible explanation for this is that tragic events need less verbal interaction and verbal space than, for instance, comic scenes; this could also be the case for 'war' and 'peace' split;
2. Tolstoy's 'war narrative' is very individualistic [Morson, 1987, p. 99], it is largely focused on the inner state of a single person on the battlefield (e.g. Andrey in the Austerlitz battle; Nikolay during the Battle of Schöngrabern and the affair at Ostróvna, Pierre in Borodino).

Therefore our hypothesis was that there should be a strong correlation between the type of settings and certain standard network metrics which reflect the intensity of interactions. The metrics we propose are:

1. network density, which is the ratio of the number of edges in a graph to the maximum possible number of edges in that graph (i.e. if each node was connected to every other node).
2. network diameter, which is the length of the longest path between one node and another in that network, measured as the number of edges. Can only be calculated if there is one single component in the graph.
3. average degree of a node (weighted and unweighted), which is also among the metrics [Elson et al., 2010] use as it shows how many connections a node (i.e. a single character) has on average in this network.

For further information on metrics we suggest fundamental work by [Wasserman, 1994].

[3] Supposedly influenced by Proudhon's *La Guerre et la Paix*, see [Eikhenbaum, 2009 (1931), pp. 498–513]

4. Networks Extraction

In literary networks nodes usually represent characters[4], while edges (and their weights) define some sort of connection or interaction between them. To build a network, one must first formalize this connection somehow. Below we list some of the most common formalizations:

1. Character co-occurrence at certain length. We assume there is an edge between two character nodes if they appear together within the same sentence or paragraph or chapter or simply a text window of a given length. This is the most primitive and abstract formalization, which is nevertheless widely used due to its simplicity. The number of cooccurrences usually becomes the weight of an edge between the characters.
2. Kinship, friendship or any other relations. Explicit mentions of relations in the text are usually quite sparse, and often it makes more sense to build such networks manually. The biggest drawback is that there are usually no weights on the edges, as the relations are mostly binary (relative or not, parent or not, spouse or not). Relation networks usually turn out relatively small and fine-grained, thus limiting the applicability of network measurements.
3. Conversational networks. The two characters are linked each time they engage in a conversation with each other, and the number or length of such conversations becomes the weight of an edge. A more sophisticated subtype of this formalization extends beyond dialogue and accounts for other sorts of social events and interactions as well (one character seeing another, characters engaging in a conflict etc.).

In our work we did not attempt to extract a complex conversational (or social events) network automatically, as this task requires very complicated processing of dialogues and identifying speakers and addressees, who are usually implicit rather than explicit in fiction (for example, [Hee et al., 2013] report that only about 25% of all speech utterances in Jane Austen feature an explicitly named speaker, while 15% have anaphoric reference to the speaker and the remaining 60% are just direct speech with no speaker mentioned at all). In addition to that, in Russian fiction speech instances are often formally indistinguishable from narrative text, as there are no quotations which could serve as formal boundaries. Given all that, we decided to markup interactions between characters in several dozen chapters of Tolstoy's novel by hand (we only marked obvious interactions), and then used these handcrafted conversational networks to evaluate character graphs that we extracted automatically using much more simplistic formalizations of interaction.

Our first set of automatically extracted networks was built on simple co-occurrence of characters in the same sentence. For the second set of networks we tried our own approach based on syntactic structures. The two characters were linked by an edge if they were both syntactic arguments under the same predicate or appeared as two conjuncts (we'll further refer to them as 'syntactic siblings'). It was

[4] Though sometimes locations are also added as separate nodes, see for example [Lee, Yeung, 2012]

our hope that this way we can filter out many 'trashy' connections inevitably made by plain co-occurrence, while still capturing many actual interactions and connections between characters, such as those expressed in examples below:

1) Обедало *человек двадцать, в том числе* **Долохов** и **Денисов.**
2) ***он*** *[Николай] вызвал* ***Наташу*** *и спросил, что такое*
3) *Il faut que vous sachiez que c'est une femme, — сказал* ***Андрей Пьеру****.*
4) *Это были* ***Наташа*** *с* ***Соней*** *и* ***Петей****, которые пришли наведаться, не встал ли.*
5) *— Голубчик, Денисов! — взвизгнула* ***Наташа****, не помнившая себя от восторга, подскочила к нему, обняла и поцеловала* ***его****.*

For our experiments we created individual networks for each of the 361 chapter of the novel, as well as bigger and denser aggregated networks for the entire parts (a part in *War and peace* may contain from 12 to 39 chapters). Here is the example of the co-occurrence network for the second part of the first volume of the novel (node sizes proportional to their weighted degrees):

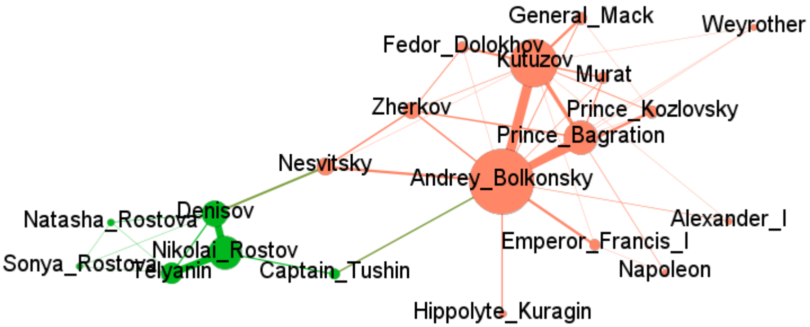


Figure 1. Co-occurrence network for the second part of the first volume of *War and peace*

Of course, before one can extract any kind of a network, character mentions themselves need to be identified throughout the text. This is a challenging task on its own, as it requires named entity recognition (NER), pronominal anaphora resolution and sophisticated nominal coreference resolution (CR). For this task we used a custom extraction model within ABBYY InfoExtractor framework [Stepanova et al., 2016]. This particular model was designed specifically for the task and had a list of *War and peace* character names and aliases. It is important to note that providing the extraction tool with character aliases is also a common practice in digital literary studies of this sort (see, for example, [He et al., 2013]), because so far, no universal NER or CR solution or tool is capable of extracting and linking characters from a random novel with tolerable quality without prior adjustment. This, of course, raises the question of scalability, especially since earlier we claimed that network analysis can be part of a large-scale 'distant reading' approach. We made an attempt to address this issue in the Conclusion and discussion section of this paper.

5. Networks evaluation

5.1. Qualitative evaluation

Before we attempted any quantitative evaluation of the networks, we chose to visualize a number of them to see if they at least 'make sense' at the first sight to someone familiar with *War and peace*. Figures 2 and 3 demonstrate two automatically extracted networks for the entire first part (first 25 chapters) of the first volume of the novel. Here the size of a node is proportional to the weighted degree of that node, and the thickness of an edge reflects its weight, i.e. frequency of co-occurrences in the same sentence or under the same predicate respectively.

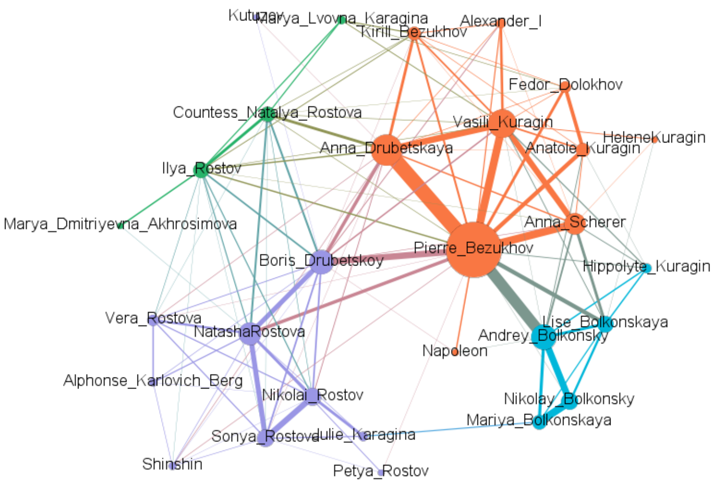


Figure 2. Co-occurrence network, first part of the first volume of *War and peace*

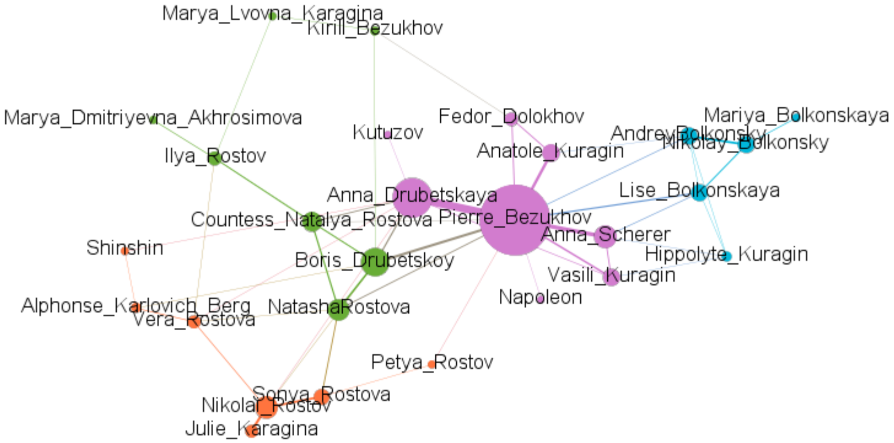


Figure 3. 'Syntactic siblings' network, first part of the first volume of *War and peace*

To a naked eye, both networks look quite similar and seem a pretty adequate reflection of the character system in the first part of the novel. One can easily see that it is centered around Pierre, who appears first at the Anna Sherer's soiree, and then becomes the center of intrigues of Vasili Kuragin and Anna Drubetskaya fighting over the legacy of Pierre's father, count Kirill Bezukhov (and after Vasili Kuragin loses this battle, he strikes back by cajoling Pierre into marrying his daughter Helene). The Rostov family is perfectly visible and, along with their Moscow nobility circle is visibly detached from the St.-Petersburg beau monde which makes up Sherer's soirees. The coloring of the pictures actually reflects automatic modularity clustering made with help of Louvain algorithm [Blondel et al., 2010], and the meaningfulness of the clusters (produced without any adjustments of the default resolution parameters) could also be a sign that the networks reflect certain information about the system of the characters. On Figure 2 one can see four automatically identified clusters:

1. the orange one is mostly St.-Petersburg *beau monde*, which at this point incorporates Pierre, once he turns from a bastard to the new count Bezukhov;
2. the purple one is mainly children and adolescents, the younger generation of Rostov family and Boris (who is, of course, a part of the "Rostov world" at this point in the plot, although already visibly leaning towards the *beau monde* cluster)
3. the turquoise one is the Bolkonsky family, and also Hyppolyte due to his repeated attempts to flirt with Lise. Doing modularity clustering with higher resolution (lesser number of clusters) would connect Andrey, Lise and Hypolyte to the *beau monde* as well (see Fig.4), while leaving Nikolay Bolkonsky and Mariya Bolkonskaya in their own Bald Hills (Лысые горы) cluster.
4. the green one is the older generation of the Rostov family and their Moscow acquaintances. If we go for lesser clusters (Fig. 4), this one merges with the younger Rostov group.

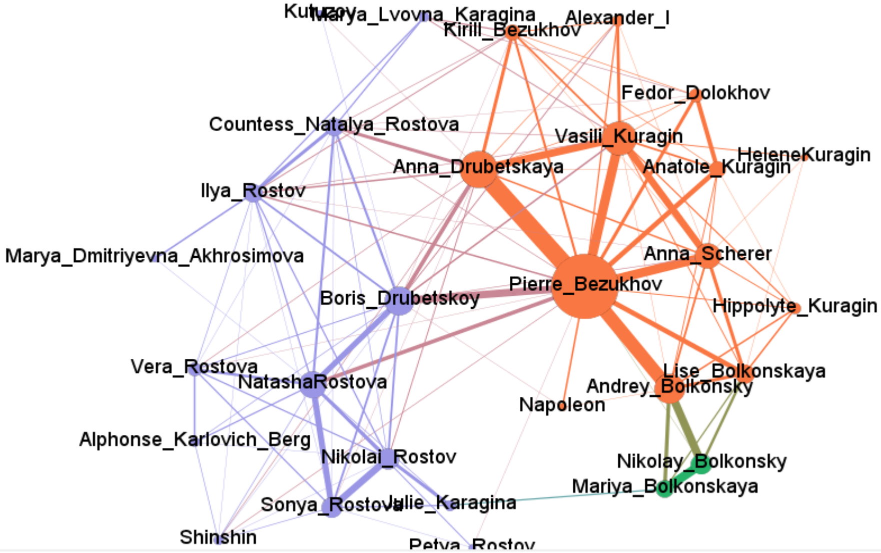


Figure 4. Co-occurrence network, first part of the first volume of *War and peace*

Unlike the “peaceful” first part of the first volume, its second part mainly takes place in the military settings, as the reader follows the experiences of Andrey Bolkonsky at the army headquarters and of Nikolay Rostov in Denisov’s squadron. The networks here (Fig. 5, Fig. 6) are visibly different from Fig. 1–2, not only in their sets of characters, but also in size, density and structure:

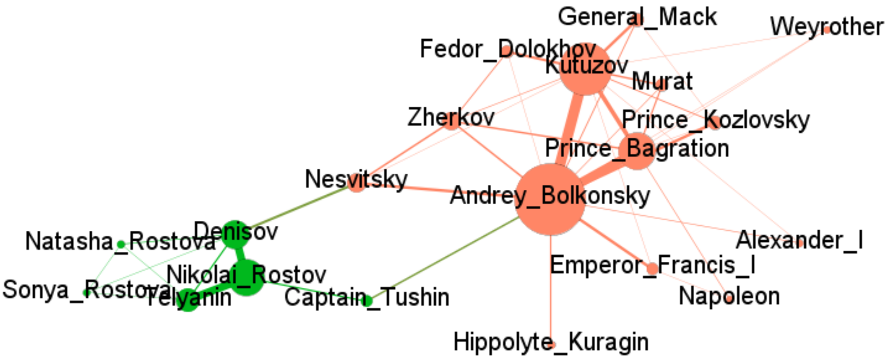


Figure 5. Co-occurrence network, second part of the first volume of *War and peace*

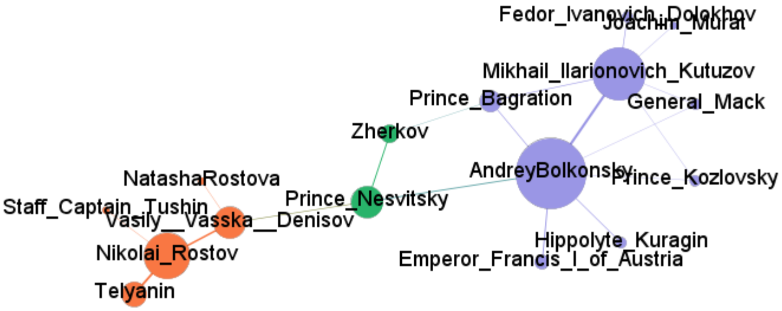


Figure 6. ‘Syntactic siblings’ network, second part of the first volume of *War and peace*

In section 4 of this paper we will try and measure these differences and find out if it occurs on a regular basis between ‘wartime’ and ‘peaceful’ parts of the novel.

The third part of the first volume is essentially a mixture of ‘war’ and ‘peace’ settings (see table 2 in the next section). Pierre spends time with the Kuragin family in St. Petersburg and eventually gets maneuvered into a marriage with Helene, prince Vasily and Anatole pay an unsuccessful visit to Bolkonsky family in the Bald Hills, while Andrey and Nikolay take part in the War of the Third Coalition and both fight in the Austerlitz battle. This heterogeneity and easily distinguishable spatial clusters of part 3 are clearly visible once we plot the graph (again, without any manual adjustments) — see Fig. 7, Fig. 8.

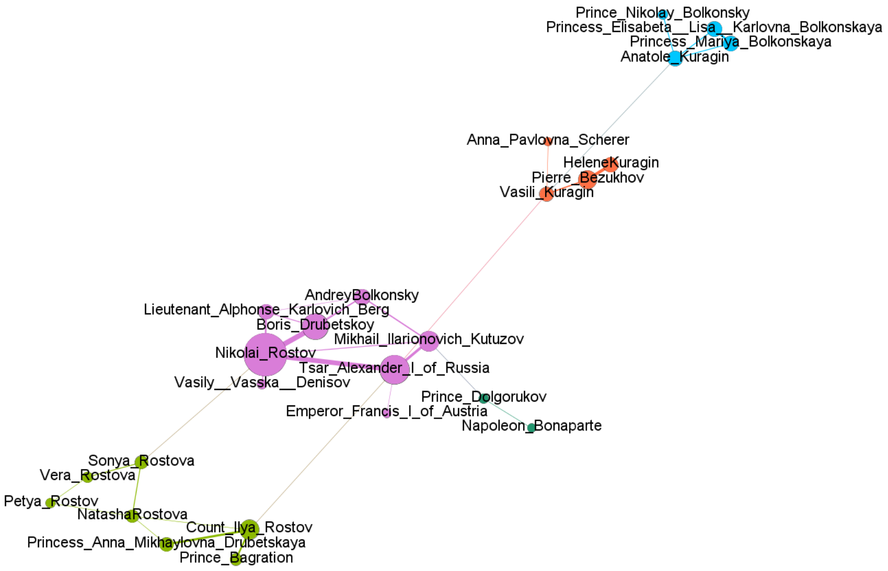


Figure 7. 'Syntactic siblings' network, third part of the first volume of *War and peace*

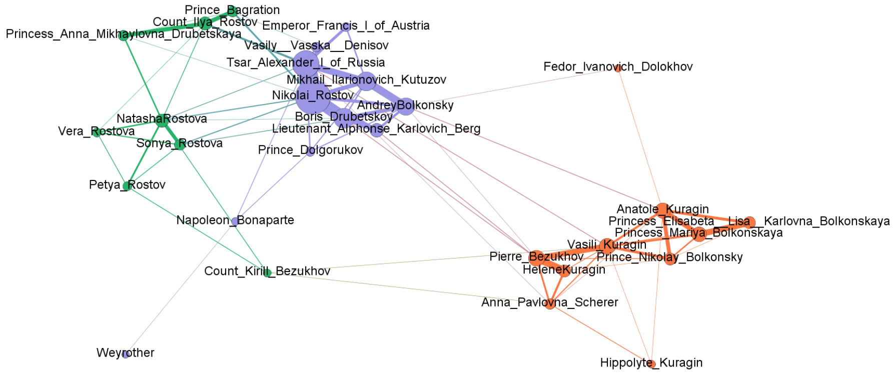


Figure 8. Co-occurrence network, third part of the first volume of *War and peace*

5.2. Quantitative evaluation

We cannot claim that our automatically extracted networks are 'meaningful' and accurately reflect the interactions of characters just because they look like it at the first sight. Therefore we also evaluate them against manually constructed interaction networks for several dozen chapters of *War and peace* (handcrafted networks available at https://github.com/DanilSko/tolstoy/tree/master/Networks/WaP_interactions).

Using Pearson correlation coefficient, we checked which network has more correlation in structural properties to the manually created one across all the chapters. As the results in Table 1 suggest, our ’Syntactic siblings’ network outperforms the standard co-occurrence one (the latter in fact has negative correlation in some cases).

Table 1. Correlation of network parameters

Parameter	Correlation with co-occurrence network	Correlation with ‘syntactic siblings’ network
Density	−0.126	0.840
Diameter	−0.456	0.219
Average degree	0.748	0.923

6. War or peace: testing the hypothesis

As the results of our evaluation suggest, the ‘syntactic siblings’ network is a much closer approximation of character interaction in a novel than the standard co-occurrence network. Now that we have chosen the method of network extraction, we can finally use it to test our hypothesis. As mentioned above, the idea was to look for correlations between certain parameters of the network and see if they correlate to the kind of settings (army/battlefield or peaceful environments, such as family or high society). We manually classified all 15 parts (excluding the epilogue, which is largely a philosophical essay) of the novel into ‘war’ (0), ‘peace’ (1) and ‘a mixture of both’ (0.5). The results of this classification are shown in the Table 2:

Table 2. Manual classification of ‘wartime’ and ‘peaceful’ parts of Tolstoy’s *War and peace*

Volume	1	1	1	2	2	2	2	2	3	3	3	4	4	4	4
Part	1	2	3	1	2	3	4	5	1	2	3	1	2	3	4
Peace/ War	1	0	0.5	1	1	1	1	1	0	0	0	0.5	0	0	0

Table 3 shows the parameters of the ‘syntactic siblings’ network for each part.

Table 3. Parameters of the ‘syntactic siblings’ network for each part of the novel

Volume	1	1	1	2	2	2	2	2	3	3	3	4	4	4	4
Part	1	2	3	1	2	3	4	5	1	2	3	1	2	3	4
Peace/ War	1	0	0,5	1	1	1	1	1	0	0	0	0,5	0	0	0
Density	0.15	0.16	0.11	0.31	0.25	0.24	0.36	0.21	0.17	0.13	0.13	0.13	0.14	0.18	0.18
Average degree	3.85	2.38	2.64	4.00	2.55	2.67	2.50	3.29	2.00	2.57	2.32	2.00	1.50	1.60	1.60
Average weighted degree	11.41	5.63	7.44	12.86	7.82	6.50	13.00	10.35	5.38	4.76	4.42	3.88	1.67	9.40	5.00

Now we can calculate correlations between the parameters of the 'syntactic siblings' networks extracted from for each part of the novel and the corresponding 'war or peace' value. Table 3 shows the resulting Pearson correlation coefficients.

Table 4. Pearson correlation coefficients between "syntactic siblings" network parameters (first column) and numeric 'war or peace' value we assigned to each part. Positive coefficient means the parameter is statistically higher in 'peaceful' parts

Parameter	Correlation with 'war or peace' value
Density	0.650
Diameter	−0.533
Average degree	0.730
Average weighted degree	0.714

As we can see from Table 4, all parameters have statistically significant correlation with our 'war or peace' target value. Density of the network and its average weighted and unweighted degrees have strong positive correlation with 'peace', quite as our hypothesis predicted. Diameter has a moderate negative correlation with the value, which means that networks with bigger diameter are more likely to be in 'war'-labeled parts.

7. Beyond 'War/peace' antithesis: more prospects for network-based research

The networks we present in this paper were originally created to calculate certain metrics and try to get a quantitative ground for a specific hypothesis. However, they can be used as a novel data-driven model for other research concerning individual characters and their relations to each other in *War and peace*. Some prospects for such research are already foreseeable from the data and visualizations we already have. They are:

- Shifts of the point of view (POV) from one character to another. This is an especially important dimension in research on *War and peace*, as its Tolstoy's trademark technique to show the unfolding events through the eyes and minds of different, constantly shifting characters [Uspensky, 1983], [Bocharov, 1971]. Nodes in the graphs have different sorts of degree and centrality measures which can be used to study he POV changes. Even if we compare our sample graphs for the three parts of the first volume of the (figures 2–8, node sizes proportional to weighted degree), we can see that the central position — and supposedly reader's main viewpoint — is taken first by Pierre, then by Andrey and finally by earlier unimportant Nikolay. Unweighted degree and betweenness centrality of nodes show similar results. Such degree and centrality changes align well with the fact that *War and peace*, when read for the first time by the contemporaries, was often initially perceived as "a novel without main heroes" [Morson, 1987, p.57]. In our networks Natasha becomes central no earlier than the second volume.

- Character groupings. Family unions (the Rostovs, the Bolkonskys, the Kuragins) with all the relations, contrasts and conflicts between them play an extremely important role in *War and peace* [Bocharov, 1971][5]. Relevant to this is the spatial opposition of Moscow vs St. Petersburg circles in 'peaceful' parts and army vs high command in 'war' parts. Without any manual adjustments, our graphs obviously cluster into these groupings (see figures 2–8 again).

8. Conclusion and discussion

We tested two rather simplistic methods for relatively 'low-cost' automatic network extraction from fictional texts and found that the approach based on syntactic structures yields results much closer to manually annotated character interaction networks.

We then used this approach to extract networks from each part of Tolstoy's *War and peace* and test our literary hypothesis. The hypothesis was that 'wartime' parts contain less intensive interaction, which can be approximated by lesser graph densities and average node degrees, as well as bigger diameters. Although all our measurements support the hypothesis, we suggest further, more substantial research before any firm arguments can be made in relation to the composition of the book and authorial techniques behind it.

Apart from this attempt to quantify the differences between 'war' and 'peace' in Tolstoy's novel, our research has revealed other potential applications of network analysis. Namely, we showed that the networks we created could provide insight on POV changes in the narrative and on character groupings and relations.

Another thing that calls for further investigation is the scalability of the approach. One may point out that if a quantitative method is being applied to just one text, however big it is, such method cannot yet be considered successful. But while we admit certain amount of manual work[6], there are two arguments for the general applicability of the approach:

1. There are no fundamental barriers for automating the whole procedure via building and adjusting NER and coreference resolution tools. The fact that state-of-the-art NER/IE/CR applications do not allow seamless transition to XIX century fiction does not imply these texts cannot be handled on a large scale in the nearest future.
2. As more and more semantically and structurally marked digital editions emerge, digital literary scholars eventually become spared from the necessity to process texts with sophisticated NLP machinery. One example is the TEI-encoded corpus of German dramatic texts used by [Trilcke et al., 2015a] for their large-scale (500 texts) network analysis. The XML markup with all the speakers tagged and identified allows easy and reproducible network creation on the go. Currently similar efforts are being made to prepare a TEI edition of Leo Tolstoy's complete works [Skorinkin, 2017].

[5] "In *War and peace* family unions, the 'breed' of the character matter a lot. In fact, the Bolkonskys and the Rostovs are more than families — they are separate modes of life" [Bocharov, 1971]

[6] Mainly the adjustment of character list initially obtained from Wikipedia

Acknowledgements

The article was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2017 (grant № 17-05-0054) and by the Russian Academic Excellence Project «5-100».

Thy research was partially supported by grant 15-06-99523 from the Russian Foundation for Basic Research.

References

1. *Agarwal A., Kotalwar A., Rambow O.* (2013), Automatic Extraction of Social Networks from Literary Text: A Case Study on Alice in Wonderland, Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan.
2. *Agarwal A., Corvalan A., Jensen J., Rambow O.* (2012), Social network analysis of Alice in Wonderland. Proceedings of the NAACL HLT 2012 Workshop on Computational Linguistics for Literature, pages 88–96, Montreal, Canada.
3. *Alberich, R., Miro-Julia, J., Rossello, F.* (2002), Marvel universe looks almost like a real social network. Preprint arXiv:cond-mat/0202174.
4. *Bocharov S.* (1971), L. N. Tolstoy's War and Peace ["Voina i mir" L. N. Tolstogo], Three masterpieces of Russian classical literature [Tri shedevra russkoi klassiki], Moscow, pp. 7–103.
5. *Bodrova, A., Bocharov, V.,* (2014), Relationship Extraction from Literary Fiction. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2014" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2014"], Bekasovo, available at: http://www.dialog-21.ru/digests/dialog2014/materials/pdf/BodrovaAABocharovVV.pdf
6. *Blondel V. D., Guillaume J., Lambiotte R., Lefebvre E.,* (2008), Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment (10), p. 1000
7. *Eikhenbaum, B.* (2009), Works on Leo Tolstoy. Saint-Petersburg. SPBSU Faculty of Philology and Arts.
8. *Elson, D. K., Dames, N. and McKeown, K.* (2010), Extracting Social Networks from Literary Fiction, Proceedings of ACL 2010, Uppsala, Sweden.
9. *Jockers M.* (2013), Macroanalysis: Digital Methods and Literary History (Topics in the Digital Humanities). University of Illinois Press; 1st Edition.
10. *Lee J., Yeung C. Y.* (2012), Extracting Networks of People and Places from Literary Texts. Proceedings of 26th Pacific Asia Conference on Language, Information, and Computation (PACLIC). pp. 209–218
11. *Moretti F.* (2013), Distant Reading. Verso, London
12. *Moretti F.* (2007), Graphs, Maps, Trees: Abstract Models for a Literary History. Verso, London
13. *Moretti F.* (2011), Network Theory, Plot Analysis. Stanford Literary Lab Pamphlets, Stanford, CA.

14. *Morson G. S.* (1987), Hidden in Plain View: Narrative and Creative Potentials in 'War and Peace'. Stanford University Press, Stanford, CA.
15. *Park, Gyeong-Mi, Kim, Sung-Hwan* (2013), Structural Analysis on Social Network Constructed from Characters in Literature Texts, Journal of Computers, Issue 8
16. *Stepanova, M., Budnikov, E., Chelombeeva, A., Matavina P., Skorinkin, D.* (2016), Information Extraction Based on Deep Syntactic-Semantic Analysis. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2016" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2016"], Moscow, pp. 721–733
17. *Skorinkin D.* (2017), Digital Edition of the Complete Works of Leo Tolstoy. 6th AIUCD Conference Book of Abstracts. Rome. pp. 264–267.
18. *Trilcke P., Fischer F., Kampkaspar D.* (2015a), Digitale Netzwerkanalyse dramatischer Texte, in: DHd2015. Von Daten zu Erkenntnissen 23. bis 27. Graz. Book of Abstracts. Austrian Centre for Digital Humanities, 2015.
19. *Trilcke P., Fischer F., Göbel M., Kampkaspar D.* (2015b), Comedy vs. Tragedy: Network Values by Genre. Network Analysis of Dramatic Texts, available at: https://dlina.github.io/Network-Values-by-Genre/
20. *Uspensky, B.* (1983). A Poetics of Composition: The Structure of the Artistic Text and Typology of a Compositional Form. Oakland. University of California Press.
21. *Wasserman, S., and Faust, K.* (1994), Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press.
22. *Weitin T.* (2017), Scalable Reading. Zeitschrift für Literaturwissenschaft und Linguistik, Volume 47, Issue 1, pp 1–6

EVALUATION TRACKS ON PLAGIARISM DETECTION ALGORITHMS FOR THE RUSSIAN LANGUAGE

Smirnov I. (ivs@isa.ru)
Institute for Systems Analysis, FRC CSC RAS, Moscow, Russia; RUDN University, Moscow, Russia

Kuznetsova R. (kuznetsova@ap-team.ru)
Antiplagiat JSC, Moscow, Russia

Kopotev M. (mihail.kopotev@helsinki.fi)
University of Helsinki, Helsinki, Finland

Khazov A. (hazov@ap-team.ru)
Antiplagiat JSC, Moscow, Russia

Lyashevskaya O. (olesar@yandex.ru)
Higher School of Economics, Moscow, Russia; Vinogradov Institute of the Russian Language RAS, Moscow, Russia

Ivanova L. (luben92@gmail.com)
Higher School of Economics, Moscow, Russia

Kutuzov A. (andreku@ifi.uio.no)
University of Oslo, Oslo, Norway

The paper presents a methodology and preliminary results for evaluating plagiarism detection algorithms for the Russian language. We describe the goals and tasks of the PlagEvalRus workshop, dataset creation, evaluation setup, metrics, and results.

Keywords: plagiarism detection, paraphrased plagiarism, source retrieval, text alignment, evaluation workshop

1. Introduction

According to the *MLA Style Manual and Guide to Scholarly Publishing* (Modern Language Association 2008: p. 166),

> [f]orms of plagiarism include the failure to give appropriate acknowledgment when repeating another's wording or particularly apt phrase, paraphrasing another's argument, and presenting another's line of thinking.

Two types of plagiarism are usually distinguished in the scholarly literature: *literal* and *obfuscated* plagiarism (Potthast et al. 2010b: 2) and *disguised* plagiarism (Gipp 2014: 12). Bela Gipp calls these two types of plagiarism *copy & paste* and *shake & paste*. The first type involves taking someone else's text word-for-word without citation, while the second involves minor modifications in another person's words, such as varying the word order, using synonyms or "padding" (Gipp 2014: 11), again without acknowledgment. According to other researchers, the *shake & paste* technique includes insertion of additional paragraphs relevant to the subject as well as mixing paragraphs. This typically leads to a sudden change in style and may remain unnoticed by a reader. When changes in an original, unattributed text are more significant (e.g., a text is paraphrased or translated), plagiarism is described as *obscured*. In paraphrasing, the source texts are reworked with the use of different linguistic tricks such as removal, word replacement, synonym substitution, word order modification, grammatical changes, and patchwriting (i.e., combining fragments from several texts) (Oakes 2014: 60). The nature of these changes depends on whether the paraphrase has occurred through manual text editing or by using automatic methods (Gupta et al. 2011: 1). For example, a manually rewritten text may be better adapted to a plagiarist's personal style than one edited automatically. Still, another case of paraphrasing is *interlingual* plagiarism, when a text is "paraphrased," in a sense, from one source language to another one. The process may include either manual or automatic translation. In the latter, an output of the machine translator usually goes through editing afterwards and obfuscation, which makes comparing the sources with the plagiarized text substantially more difficult while at the same time showing evidence of translation.

In the academic community, the problem is especially crucial in connection with student papers and popular scientific literature. Plagiarism is especially difficult to define in the latter case, since such literature describes facts that are already known and often cannot be reformulated differently. Thus, establishing both the evidence for and the limits of plagiarism becomes more challenging and problematic. In contrast, student plagiarism usually can be detected using basic automated tools. Its widespread occurrence today is primarily the result of the tolerance on the part of educators and the academic community, which makes plagiarism a common practice. In 2004, for instance, it was estimated that 10 percent of student works in the United States and Australia involved plagiarism (Oakes 2014: 60). In more recent research, 36 percent of respondents in Russia admitted to regularly copying the texts of others in different forms (Kicherova et al. 2013: 2). According to a study conducted in 2013 (Maloshonok 2016), as many as 36.7 percent of undergraduate students in eight Russian universities take personal credit for works they have downloaded from the Internet. However, the problem is not limited to students' activity. In 2011 in Germany, two cases of plagiarism were documented in Ph.D. dissertations. Those cases were analyzed in detail by the GuttenPlag community and provided the basis for the monograph *False Feathers: A Perspective on Academic Plagiarism* (Weber-Wulff 2014: 29). In Russia, the same problem has been diagnosed by the Dissernet grassroots movement (www.dissernet.org), whose purpose is to reveal plagiarism in scientific texts (see Golunov 2014; Denisova-Schmidt 2016).

As disguising plagiarism becomes more and more sophisticated, detecting it requires newer and more advanced techniques. At the moment, there are several services that are able to detect plagiarism in Russian-language texts (see Nikitov et al.

2012), but thus far there has been no systematic evaluation of these services. This paper and the PlagEvalRus workshop it stems from are the first attempts to define the problem of how to evaluate plagiarism and outline ways of handling it.

There are several related workshops and events on similarity detection on both word and sentence levels. The Russian language is a primary target for two of them: 1) RUSSE (Panchenko et al. 2015), the shared task on word-level semantic similarity; and 2) ParaPhrase (Pivovarova et al. 2016), the shared task on sentence-level paraphrase detection, i.e. identification of sentences that have similar meaning but not necessarily similar in structure. The series of related workshops, SemEval, includes a task on Semantic Textual Similarity (Agirre et al. 2016), which is aimed to measure degree of semantic similarity between two text snippets, written in English and some other languages (but not in Russian). However, in plagiarism detection tasks, snippets of reused texts are not given, but supposed to be retrieved from source texts, thus this task is significantly more complicated to accomplish. The most closely related to the PlagEvalRus seminar are PAN workshops (e.g. Potthast et al. 2010a) that have several tasks on plagiarism detection.

2. Goals and tasks

In this article the methodology we propose for detecting plagiarism in the Russian language is based on years of experience of the PAN network (a series of events on digital text forensics [e.g., Potthast et al. 2010a, Potthast et al. 2010b, 2014]; see more on http://pan.webis.de). We have focused on evaluating algorithms oriented toward monolingual Russian plagiarism with an emphasis on scientific texts (academic plagiarism). In our workshop, called PlagEvalRus and held during 2016-2017, we offered the following tracks after holding preliminary discussion:

- Track 1: Plagiarized sources retrieval
- Track 2: Copy and paste plagiarism detection
- Track 3: Paraphrased plagiarism detection.

Track 1 corresponds to the Source Retrieval (SR) task evaluated at the PAN competitions. The participants received a dataset, which includes potential sources and suspicious texts; the latter contained both literal and paraphrased plagiarism. The participants are required to provide a list of sources for each suspicious text (more details below), sorted according to the number of reused fragments in descending order; unlike the PAN Source Retrieval task does not require any sorting of detected text pieces. Track 1 was thus quite similar to the search tracks on the Russian Information Retrieval Evaluation Seminar (see http://romip.ru/en), the difference being that the search queries in our case were much longer textual excerpts.

Tracks 2 and 3 entirely correspond to the Text Alignment (TA) task evaluated at the PAN competitions; i.e., in a pair of texts given to participants, fragments taken from one text need to be found in a second text. A **fragment** is a sequence of at least five tokens excluding stop words. **Literal reusing** means a full correspondence of character strings ignoring blank and hidden characters. **Paraphrased reusing** is rewriting the original text preserving the idea of a reused fragment. Thus, Track 2 was intended to detect literal plagiarism, while Track 3 involved detecting illicit paraphrasing.

3. Dataset

For each track, the organizers provided two datasets, training and testing, along with a text collection that contains, among other things, potential sources. Participants were supposed to train their algorithms on the training dataset, which was provided to all participants and could be read on the Workshop's site, www.dialog-21.ru/en/evaluation/2017/plageval, well in advance. The participants received clear instructions on how to handle the data. All scripts, datasets and instructions are freely accessible at https://plagevalrus.github.io.

3.1. Collection of sources

The "potential sources" dataset contains about 5.7 million Russian texts, compiled from the following resources:

- Russian Wikipedia: about 1.3 million texts;
- Student essays from open online collections: about 3.3 million texts;
- Open-sourced book-sized academic texts: about 12,000 texts;
- Academic papers from the open access resource Cyberleninka.ru: 1 million texts.

All texts were converted to the plain-text format in UTF-8. Evident duplicates were preliminarily removed, and the remaining files were then mixed. Each text was stored in a separate file with a name containing a unique identifier.

3.2. Suspicious Texts

The test dataset was created under the same conditions as the training dataset. In line with the PAN workshops (Potthast et al. 2010a), the following types of texts were specified:

1) **Automatically generated copy and paste plagiarism**. To do this, we randomly selected sentences from a target text and changed each of them by one or more randomly chosen consecutive sentences from source texts, which did not belong to the target collection. Each fragment was identified by its beginning and its length in characters.

   The resulting target texts contain from 10 to 80 percent of plagiarized material (calculated in sentences).

2) **Automatically generated paraphrased plagiarism**. The collection containing this type of reused texts was created in the same way as the copy and paste texts, except that the sentences of the source texts were automatically paraphrased by using one or more of the following methods:
   - Replacing words with their synonyms;
   - Adding and removing synonym chains;
   - Abbreviation and amplification;
   - Adding and removing diminutives;
   - Singular/plural replacement.

   For a detailed description of the procedure, see (Khazov and Kuznetsova 2017).

3) **Manually generated copy and paste plagiarism**. This dataset was compiled from academic texts, the sources of which are known and available on the Internet. The texts with the manually created word-for-word fragments were used only for Track 1.

4) **Manually paraphrased plagiarism**. Compiling such a collection was motivated by the activity of those "authors" who reuse fragments from various sources trying to obfuscate their borrowings by paraphrasing. This collection is built of essays reflected different topics; creators were instructed to select a text from the source collection, to mark and paraphrase fragments, and then to insert them into a Microsoft Excel table. The procedure like this makes it possible to extract the markups and transform it into different tasks related to a plagiarism detection evaluation. A fragment that has been restructured should contain at least one sentence. The creators were allowed mixing sentences from different sources and inserting original sentences between plagiarized ones. Therefore, the resulted essays contain both original and paraphrased fragments, which are produced by creators under the following condictions:
   - deleting words (about 20%) from an original sentence;
   - adding words (about 20%) into an original sentence;
   - replacing words or phrases with synonyms, reordering clauses, adding new words, changing word forms (number, case, form and verb tense, etc.); about 30% in an original sentence;
   - changing the order of words or clauses in an original sentence;
   - concatenating two or more original sentences into one;
   - splitting an original sentence into two or more (with a possible changing in order of how they appear in a text);
   - replacing words or phrases of an original sentence with synonyms (e.g. "sodium chloride" → "salt"), replacing abbreviations to their full transcripts and vice versa, replacing personal names with their initials, etc.;
   - complex rewriting of an original sentence, which combines 3-5 or more aforementioned techniques. This type involves significant changes in a source text by paraphrasing idioms, synonymic modification of structures, permutation of words or parts of a complex sentence, etc. Using this technique effectively produces paraphrased texts: in some cases even experts could hardly consider the rewritten text as plagiarized;
   - coping a sentence from a source and pasting it into an essay with no significant changes.

Therefore, each essay contains no fewer than 100 paraphrased sentences, 90 percent of the texts being taken from at least five sources. For a detailed description of the procedure, see (Sochenkov et al. 2017). Table 1 shows the number of texts and pairs <suspicious text, source text> in both training and the testing data.

Table 1. The Training and the Test Data sets: size in the number of texts and pairs

	Training set		Test set		
	Texts for SR and TA	Pairs	Texts for SR	Texts for TA	Pairs
Automatically generated copy&paste plagiarism	1,000	4,257	5,000	100	268
Automatically generated paraphrased plagiarism	2,000	4,251	5,000	100	297
Manually copy&paste plagiarism	519	—	519	—	—
Manually paraphrased plagiarism	152	913	38	39	234
Total	3,671	9,421	10,557	239	799

Figure 1 shows the texts we suspected of plagiarizing (from 1 to 19 sources).

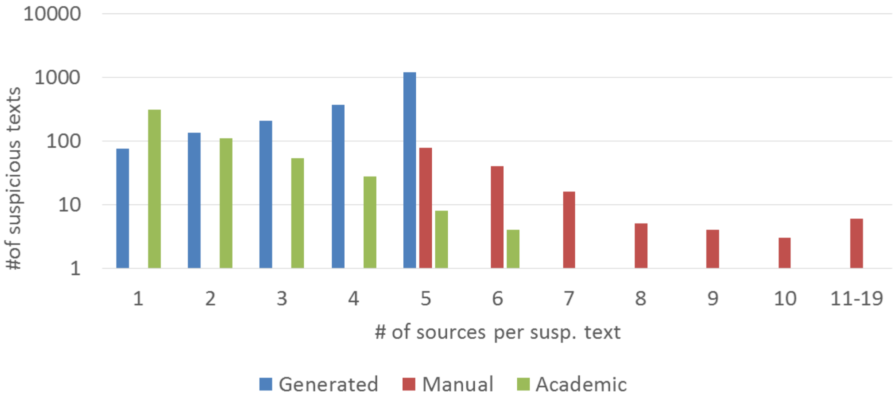


Figure 1. Texts suspected of plagiarizing N sources (where N ranges from 1 to 19)

4. Evaluation

4.1. Evaluation Setup

The evaluation of the results on Track 1, Source Retrieval, differs significantly from that on Tracks 2 and 3, Text Alignment. On Track 1, the participants downloaded the collection of sources and searched for copied fragments using a system of the participant's own devising. The result is supposed to be a list containing sources for each suspicious text, ranked (in descending order) according to the number of fragments detected. Those following this track were asked to deliver results for a maximum of 5 runs. In evaluating the runs, the participants' responses were automatically evaluated against the benchmark created by the PlagEvalRus Workshop's organizers. A baseline was not offered for the source retrieval track due to both the complexity of the task and lacking time needed for it development.

For Tracks 2 and 3, plagiarism is considered successfully detected if a fragment found by a system is located or completely within a text marked as such in the test collection. Coincidences in texts were not taken into account. Therefore, any fragment detected, but not marked in the test collection was not registered for the evaluation. The PAN baseline method was used in comparing results. This brute method is based on a simple shingles approach with chunks of 50 symbols length.

To evaluate the systems on Tracks 2 and 3, we used the TIRA platform (http://www.tira.io),1 which ensured reproducibility and neutrality in evaluating the algorithms. Each participant in Track 2 or 3 was provided with a virtual machine on the TIRA server in order to run his/her system on a given test set. The evaluation was performed automatically on the server and the results were available to the participant. The overall results are available only to the administrator of the TIRA service.

4.2. Evaluation Metrics

4.2.1. Source Retrieval

Let T_{src} denote a set of source texts for suspicious text t_{plg}, and let T_{ret} denote the set of texts that is retrieved by a source retrieval algorithm when given t_{plg}. Then precision (P) is defined as

$$P = \frac{|T_{ret} \cap T_{src}|}{|T_{ret}|}$$

and recall (R) as

$$R = \frac{|T_{ret} \cap T_{src}|}{|T_{src}|}$$

The PAN metrics (Potthast et al. 2014) measures the effect of near-duplicate web documents, but we do not take into account similar texts from T_{ret}. Furthermore, full duplicates were preliminarily removed from the collection of sources.

We define F-measure (F) as

$$F = \frac{2 * R * P}{R + P}$$

The results of Track 1 were supposed to be ranked in descending order according to number of reused fragments detected, so that we could assess the quality of ranking. The text t_{ret} is relevant to t_{plg} if $t_{plg} \in T_{ret} \cap T_{src}$. Precision at k ($P@k$) is a measure of ranking performance for t_{plg} and is defined as the number of relevant texts among the first k retrieved results, divided by k.

The average precision (AP) for t_{plg} is the average of $P@k$ for all relevant texts:

[1] TIRA is currently one of the few platforms (if not the exclusive one) that support software submissions with a little extra effort; it has been utilized for several similar shared tasks within PAN@CLEF, CoNLL, etc.

$$AP(t) = \frac{1}{|K|}\sum_{k\in K} P@k$$

where *K* stands for a set of positions of all relevant texts. The mean average precision (*MAP*) is the mean of the average precision for each text from a set of suspicious texts denoted T_{plg}.

$$MAP = \frac{1}{|T_{plg}|}\sum_{t_{plg}\in T_{plg}} AP(t_{plg})$$

4.2.2. Text Alignment

Following (Potthast et al. 2010b), let *S* denote the set of plagiarism cases in the corpus, and let *R* denote the set of detections reported by a plagiarism detector for the suspicious documents. A plagiarism case $s = \langle s_{plg}, d_{plg}, s_{src}, d_{src}\rangle$, $s \in S$, is represented as a set *s* of references to the characters of t_{plg} and t_{src}, specifying the passages s_{plg} and s_{src}. Likewise, a plagiarism detection $r \in R$ is represented as *r*. Based on this notation, both macro- and micro-averaged precision and recall of *R* under *S* can be measured as follows:

$$precision_{micro}(S,R) = \frac{\left|\bigcup_{(s,r)\in(S\times R)} (s \sqcap r)\right|}{\left|\bigcup_{r\in R} r\right|}$$

$$precision_{macro}(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{\left|\bigcup_{s\in S} (s \sqcap r)\right|}{|r|}$$

$$recall_{micro}(S,R) = \frac{\left|\bigcup_{(s,r)\in(S\times R)} (s \sqcap r)\right|}{\left|\bigcup_{s\in S} s\right|}$$

$$recall_{macro}(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{\left|\bigcup_{r\in R} (s \sqcap r)\right|}{|s|}$$

where $$s \sqcap r = \begin{cases} s \cap r \text{ if } r \text{ detects } s, \\ \quad \emptyset \quad \text{otherwise.}\end{cases}$$

The macro-averaged variants are allotted equal weight in each plagiarized case, regardless of length. Conversely, the micro-averaged variants favor detecting long plagiarized fragments, which are generally easier to identify.

To address the fact that plagiarism detectors sometimes reported overlapping or multiple detections for a single plagiarism case, let a detector's granularity be defined as:

$$granularity(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_S|$$

where $S_R \subseteq S$ are cases detected by detections in *R*, and $R_S \subseteq R$ are detections of *s*; i.e., S_R $\{s|s \in S \wedge \exists r \in R{:}r \text{ detects } s\}$ and R_S $\{r|r \in R \wedge r \in R{:}r \text{ detects } s\}$. The three above-mentioned measures taken individually do not allow single ranking based on these approaches. To make a uniform ranking, the measures are combined into a single overall score, called the Plagdet score and calculated as follows:

$$plagdet(S,R) = \frac{F_1}{\log_2(1 + gran(S,R))}$$

where F_1 is the equally weighted harmonic mean of precision and recall.

4.3. Evaluation Results

Only one team participated in all offered Tracks (Hereafter referred to as **zubarev**; see Zubarev and Sochenkov 2017). The results of all runs are shown in Tables 2–8.

4.3.1. Track 1: Plagiarized source detection

The evaluation results for the track are presented in Tables 2–4.

Table 2. Evaluation results for the automatically-generated copy and paste and paraphrased plagiarism retrieval tasks

		generated copy&paste plagiarism				**generated paraphrased plagiarism**			
team	**Run**	**MAP**	**P**	**R**	**F1**	**MAP**	**P**	**R**	**F1**
zubarev	zubarev.1	0.603	0.222	0.779	0.346	0.593	0.234	0.745	0.357
	zubarev.2	0.151	0.005	0.785	0.011	0.202	0.005	0.750	0.011

Table 3. Evaluation results for manual copy and paste and paraphrased plagiarism retrieval task

		manual copy&paste plagiarism				**manually paraphrased plagiarism**			
team	**run**	**MAP**	**P**	**R**	**F1**	**MAP**	**P**	**R**	**F1**
zubarev	zubarev.1	0.851	0.106	0.974	0.191	0.608	0.441	0.830	0.576
	zubarev.2	0.610	0.003	0.978	0.006	0.390	0.009	0.989	0.019

Table 4. Evaluation results for overall source retrieval tasks

		Total			
team	**runs**	**MAP**	**P**	**R**	**F1**
zubarev	zubarev.1	0.664	0.251	0.832	0.368
	zubarev.2	0.338	0.005	0.876	0.012

The participant has submitted 36 sources in average for each suspicious text in zubarev.1 run and 579 sources in average for each suspicious text in zubarev.2 run, so the second run is obviously optimized for higher recall. As one can see, the best F1 and MAP was gained on manual plagiarism detection. We suppose the reason behind that is a topical heterogeneity of automatically generated texts that might affect participant's algorithms. The results in general correspond to average results of PAN participants, who showed the highest F1 equaled 0.47 at PAN2015.

4.3.2. Track 2: Copy and paste plagiarism detection

The evaluation results for the automatically-generated copy and paste plagiarism retrieval are shown in Table 5.

Table 5. Evaluation results for automatically-generated copy and paste plagiarism detection. Macro- and Micro-average

team.run	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	1.0046	0.7240	0.9101	0.8038	0.9615	0.9943	0.9744
zubarev17.1	1.5084	0.9496	0.6427	0.5778	0.9828	0.8217	0.6746
zubarev17.2	1.4660	0.9320	0.7013	0.6146	0.9776	0.8588	0.7022

In this track, the PAN baseline outperforms Zubarev's detector by all measures except precision. In general, the task of copy and paste plagiarism detection has been solved well enough.

4.3.3. Track 3: Paraphrased plagiarism detection

The evaluation results for paraphrased plagiarism retrieval are shown in Tables 6–7.

Table 6. Evaluation results for automatically-generated paraphrased plagiarism detection. Macro- and Micro-average

team.run	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	3.4639	0.9051	0.6895	0.3626	0.9710	0.8334	0.4156
zubarev17.1	1.5404	0.9604	0.6730	0.5884	0.9875	0.8219	0.6670
zubarev17.2	1.4834	0.9473	0.7340	0.6303	0.9812	0.8650	0.7006

Table 7. Evaluation results for manually paraphrased plagiarism detection. Macro- and Macro-average

team.run	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	1.1414	0.8332	0.0554	0.0946	0.8960	0.0761	0.1277
zubarev17.1	1.0015	0.8068	0.3409	0.4788	0.8845	0.3815	0.5325
zubarev17.2	1.0016	0.6250	0.4715	0.5369	0.8208	0.5312	0.6443

In this track, Zubarev's detector outperforms PAN baseline by all measures. The results of generated paraphrased plagiarism detection are better than results for manually paraphrased texts, though granularity is better for the last. The reason of a granularity gap is probably connected with the difference in length of fragments in the tasks: in manually paraphrased texts, the reused fragments equal to sentence, while in automatically generated paraphrased texts, the reused fragments equal to a paragraph (up to 10 sentences).

We can see that the measures on copy and paste plagiarized texts are expectedly higher than measures on paraphrased texts almost in all cases. Nevertheless, the most complicated task of paraphrased plagiarism detection is solved by Zubarev detector quiet well while PAN baseline dropped down Recall and Plagdet in this task.

4.3.4. Plagiarism detection for both types

Evaluation results for automatically-generated copy and paste, automatically-generated and manually paraphrased plagiarism detection are shown in Table 8.

Table 8. Evaluation results for overall text alignment tasks. Macro- and Micro-average

team.run	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	1.9953	0.8525	0.3366	0.3049	0.9637	0.6893	0.5078
zubarev17.1	1.3028	**0.9129**	0.4605	0.5087	**0.9693**	0.7043	0.6780
zubarev17.2	**1.2417**	0.8158	**0.5644**	**0.5729**	0.9460	**0.7737**	**0.7309**

In the overall text alignment task, the Zubarev's detector (which is based on sentence similarity) performed by the Plagdet better than the PAN baseline (which is based on character shingles. The Zubarev's detector also performed better in all types of plagiarism except an automatically-generated copy and paste variation. In the PlagEvalRus test dataset, the PAN baseline demonstrated results comparable to those on the PAN test dataset in English (Potthast et al. 2014). Finally, we should notice that micro-measures are always higher than macro.

5. Conclusions and further advances

In this article, we have presented the methodology and the datasets for plagiarism detection evaluation algorithms in monolingual Russian texts. Owing to circumstances beyond our control, only one of all the teams which signed up for the PlagEvalRus Workshop submitted its results. Participants' feedback showed that computational complexity and lack of both high-performance computing facilities and large-scale storage systems caused no-bid decisions. Our decision to lay upon TIRA technical solutions should obviously be reconsidered in our further workshops, because the participants have had to invest much time in studying this evaluation framework. Nevertheless, the TIRA framework allows and we agreed to make the text alignment task continuously available for evaluation on the TIRA site (http://www.tira.io/tasks/pan/#text-alignment; see the dataset "pan17-text-alignment-test-dataset-dialogue17-russian-2017-02-22"), so that anyone who submits his/her software can obtain the results for comparison.

Preparation of manually paraphrased texts was the most laborious phase in any workshop like ours. According to our estimations, preparing one essay takes in average from 4 to 10 hours; the properly formed essays are not always resulted on the first try, a (semi)automated verification is always required for this time-consuming preparatory work. Taking both our experience and participants' needs into consideration, we intend to hold PlagEvalRus workshop for the second time next year. We plan to enlarge collection of sources and increase the size of training datasets. We will discuss offering a joint plagiarism detection track, where both source retrieval and text alignment are not separated. We also plan to announce a cross-language (translated) plagiarism detection track expecting more participants at our Workshop.

Acknowledgments

We would like to thank the following people and institutions for various kinds of assistance in organizing this Workshop:

- For both technical support and inspiration: Martin Potthast (PAN founder, Digital Bauhaus Lab.);
- For the data provided: Cyberleninka.ru and other institutions;
- For the preparation of datasets: students of RUDN University, students of the Higher School of Economics in Nizhny Novgorod (A. Safaryan, O. Andriyanova, N. Babkin, A. Bazyleva, A. Beloborodova, Ju. Frolova, M. Kurilina, M. Petrova, V. Rybakov, T. Semenova, A. Sorokina, T. Sharipova, A. Tryaskova, V. Vdovina) and Moscow (S. Malinovskaya, Z. Evdaeva, A. Stepanova, D. Suslova).

References

1. *Agirre E. et al.* (2016) Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation //Proceedings of SemEval. — 2016. — pp. 497–511.
2. *Gipp, B.* (2014), Citation-based Plagiarism Detection. Detecting Disguised and Cross-language Plagiarism Using Citation Pattern Analysis. Springer Fachmedien Wiesbaden. http://link.springer.com/book/10.1007%2F978-3-658-06394-8.
3. *Golunov, S.* (2014), The Elephant in the Room: Corruption and Cheating in Russian Universities. Columbia University Press.
4. *Gupta, P., Singhal, K., Majumder, P., Rosso, P.* (2011), Detection of Paraphrastic Cases of Mono-lingual and Cross-lingual Plagiarism. In Proceedings of ICON-2011, Macmillan Publishers, India. http://users.dsic.upv.es/~prosso/resources/GuptaEtAl_ICON11.pdf
5. *Denisova-Schmidt, E.* (2016), Corruption in Russian Higher Education. Russian Analytical Digest, 191: 5–9.
6. *Khazov, A., Kuznetsova, M.,* (2017) Automatic Generation of Verbatim and Paraphrased Plagiarism Corpus. In press.
7. *Kicherova, M., Kyrov, D., Smykova, P., et al.* (2013), Plagiarism in Students' Papers: Toward the Roots of the Problem [Plagiat v studencheskikh rabotakh: analiz sushhnosti problemy]. Online journal Naukovedenie (IGUPIT), 4. http://naukovedenie.ru/PDF/83pvn413.pdf

8. *Modern Language Association* (2008), MLA Style Manual and Guide to Scholarly Publishing (3rd ed.). New York: Modern Language Association of America.
9. *Maloshonok, N.* (2016), How Perception of Academic Honesty at the University Is Linked with Student Engagement: Conceptualization and Empirical Research Opportunities [Kak vospriyatie akademicheskoj chestnosti sredy universiteta vzaimosvyazano so studencheskoj vovlechennost'yu: vozmozhnosti kontseptualizatsii i ehmpiricheskogo izucheniya]. Voprosy obrazovanija, 1: 35–60.
10. *Nikitov, A., et al.* (2012), Plagiarism in Under- and Postgraduate Students' Papers: Problem and Actions Against [Plagiat v rabotakh studentov i aspirantov: problema i metody protivodejstviya]. Universitetskoe upravlenie: praktika i analiz, 5: 61–68.
11. *Oakes, M.* (2014), Literary Detective Work on the Computer. Amsterdam: John Benjamins Publishing Company.
12. *Panchenko, A., Loukachevitch, N. V., Ustalov, D., Paperno, D., Meyer, C. M. and Konstantinova, N.,* (2015) RUSSE: The First Workshop on Russian Semantic Similarity, Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue", vol. 2, pp. 89–105.
13. *Pivovarova L., Pronoza E., Yagunova E.* (2016) Shared Task on Sentence Paraphrase Detection for the Russian Language // http://www.paraphraser.ru/download/get?file_id=2
14. *Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., Rosso, P.* (2010a), Overview of the 2nd International Competition on Plagiarism Detection. Martin Braschler and Donna Harman (Eds.): Notebook Papers of CLEF 2010 LABs and Workshops, 22–23 September, Padua, Italy.
15. *Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.* (2010b), An Evaluation Framework for Plagiarism Detection. In Proceedings of the 23rd international conference on computational linguistics: Posters: 997–1005. http://www.uni-weimar.de/medien/webis/publications/papers/stein_2010p.pdf
16. *Potthast, M., Hagen, M., Beyer, A., Busse, M., Tippmann, M., Rosso, P., Stein, B.* (2014), Overview of the 6th International Competition on Plagiarism Detection. In CEUR Workshop Proceedings, 1180: 845–876. https://www.uni-weimar.de/medien/webis/publications/papers/stein_2014k.pdf
17. *Sochenkov, I., Zubarev, D., Smirnov, I.* (2017) PARAPLAG: Russian Dataset for Paraphrased Plagiarism Detection. In press.
18. *Weber-Wulff, D.* (2014), False Feathers. A Perspective on Academic Plagiarism. http://link.springer.com/book/10.1007%2F978-3-642-39961-9.
19. *Zubarev, D., Sochenkov, I.* (2017) Paraphrased plagiarism detection using sentence similarity. In press.

THE PARAPLAG: RUSSIAN DATASET FOR PARAPHRASED PLAGIARISM DETECTION

Sochenkov I. V. (sochenkov_iv@rudn.university)[1,2],
Zubarev D. V. (zubarev@isa.ru)[1,2], **Smirnov I. V.** (ivs@isa.ru)[3,1]

[1]RUDN University, Moscow, Russia; [2]Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia; [3]Institute for Systems Analysis, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

The paper presents the ParaPlag: a large text dataset in Russian to evaluate and compare quality metrics of different plagiarism detection approaches that deal with big data. The competition PlagEvalRus-2017 aimed to evaluate plagiarism detection methods uses the ParaPlag as a main dataset for source retrieval and text alignment tasks. The ParaPlag is open and available on the Web. We propose a guide for writers who want to contribute to the ParaPlag and extend it. The analysis of text rewrite techniques used by unscrupulous authors is also presented in our research.

Keywords: paraphrased plagiarism detection, text reuse detection, dataset for plagiarism detection evaluation.

PARAPLAG: КОРПУС ДЛЯ ВЫЯВЛЕНИЯ ПЕРЕФРАЗИРОВАННЫХ ТЕКСТОВЫХ ЗАИМСТВОВАНИЙ НА РУССКОМ ЯЗЫКЕ

Соченков И. В. (sochenkov_iv@rudn.university)[1,2],
Зубарев Д. В. (zubarev@isa.ru)[1,2],
Смирнов И. В. (ivs@isa.ru)[3,1]

[1]Российский университет дружбы народов, Москва, Россия; [2]Федеральный исследовательский центр «Информатика и управление» Российской академии наук, Москва, Россия; [3]Институт системного анализа, Федеральный исследовательский центр «Информатика и управление» Российской академии наук, Москва, Россия

Ключевые слова: выявление перефразированных заимствований, текстовые заимствования, оценка качества методов выявления текстовых заимствований

1. Introduction

Modern information technologies have given to unscrupulous authors a simple and effective tool to usurp someone else's results with minimal effort. Modern plagiarism detection systems (PDS), such as TurnitIn[1], Antiplagiat[2] and others detect "copy and paste" plagiarism in research or student papers with high recall and precision. Therefore unscrupulous authors use various obfuscation techniques (noise bringing) to their text documents without substantial modification of its meaning and content. This obfuscation can be done automatically by disturbing plagiarized text fragments using hidden text, pictures, non-standard symbols, formulas etc. to prevent correct text extraction from documents in common formats (PDF, Microsoft Word). PDS developers considered this problem, so most of such obfuscation techniques can be successfully detected. The other ways to hide the plagiarism from PDS is to use a little more time consuming techniques: paraphrasing the original text and/or translation from another language. In such case, it expands from "text stealing" (improper citation) to wrongful appropriation of thoughts and ideas. Therefore, the task is beyond the detection of improper citations and goes to judgment of novelty and originality of information presented in scientific or student papers. This judgment is a vital part of scientific expertise or student works checking. Obviously, such expertise is impossible without PDS. However, the detection of rewritten paraphrased and/or translated texts is challenging for modern PDS. Therefore, the research of new methods for detection of paraphrased and translated plagiarism on big data sources is an important scientific task related to information retrieval.

The best and complete solution could not be found without well-grounded and representative evaluation of different approaches for the addressed problems. The evaluation of information retrieval methods for modern PDS requires a large dataset containing original and plagiarized texts for training and testing.

This research addresses the task of creating a dataset for plagiarism detection. It focuses on paraphrased text plagiarism that comes from the real world practice. The main research goal is to create the ParaPlag—a large text dataset in Russian to evaluate and compare quality metrics of different plagiarism detection approaches that deal with a big data. The analysis of text rewrite techniques used by unscrupulous authors is also in focus in our research. We also propose a guide for writers who want to contribute to the ParaPlag and extend it. The competition PlagEvalRus-2017 aimed to evaluate quality of PDS uses the ParaPlag as a main dataset for source retrieval and text alignment tasks.

2. Related work

It is very important to have a standardized dataset to evaluate new plagiarism detection methods. Having such dataset, it is possible to get comparable results of the evaluation. The dataset should be large and represent different text reuse techniques. Therefore, one can test a plagiarism detection method in conditions, which are close to the terms of a real world.

[1] http://turnitin.com

[2] http://antiplagiat.ru

Actually there are a few open datasets for such evaluation and mostly used is PAN-PC-11 corpus (Potthast et al., 2010). This corpus was used in PAN competition that held yearly since 2009 until 2015 year. The corpus consists of texts in English that were created by borrowing text of books from Gutenberg collection. Reused text was modified automatically and manually. Since text is borrowed randomly from any book, the suspicious documents do not belong to the same topic as sources. This is the main concern related with this corpus and it makes it suitable only for evaluation of the text alignment task.

The paper (Gollub, T. et al., 2012) introduces the TIRA platform as a standard framework for PDS evaluation. It is a playground for text alignment and source retrieval tracks. TIRA contains Webis-TRC-12 (Potthast et al., 2013) that aims to address the aforementioned issue. Each suspicious document from this corpus was created manually and writers should have tried to hide the plagiarism. Web pages from ClueWeb dataset[3] are the sources for manually written essays (Potthast et al., 2013). The writers found the materials for their plagiarized essays using Chat Noir (Potthast et al., 2012) and Indri search engines. Therefore, the source documents are safely hidden among tens of millions of web pages. Thus, this corpus is suitable for source retrieval task on PAN competition. Under the TIRA this task can be solved using the aforementioned search engines as an entry point to the ClueWeb dataset. In common, suspicious passages are queries and search results are candidates for deep analysis. However the real world PDS require their own indexes and special data structures to deal with plagiarism with high efficiency. Therefore, one needs to index about 504 million web pages (the size of the part of ClueWeb in English) to deal with source retrieval and text alignment in real applications.

Other related research includes the Semeval workshop, which has a corpus for methods that estimate Semantic Textual Similarity (Agirre, E., et al., 2015). This task is close to the text alignment for plagiarism detection, but it focuses on more precise alignments between chunks. The corpus contains sentences in English and Spanish (news headlines, image descriptions, answer pairs from a tutorial dialogue system, etc.) but it is relatively small and therefore it could not be used as a dataset for complex plagiarism detection.

The competition on paraphrase detection in Russian texts uses the specially created and extensible open corpus containing sentence pairs (Pronoza, E., et al. 2016). The task follows the standard procedure: the participating method takes a pair of sentences and returns the similarity class as a response. There are three cases: precise paraphrase, near paraphrase and non-paraphrase.

Other research (Burrows, S. et al., 2013) studies some paraphrase techniques (including translation) and discusses the construction of a paraphrase corpus via crowdsourcing. It also gives a brief review for some other datasets mostly containing paraphrases at sentence-level developed in English: (Dolan, W. B., Brockett, C., 2005), (Clough, P. et al., 2002). The research by Madnani and Dorr (Madnani, N., Dorr, B. J., 2010) discusses the automatic generating of phrasal and sentential paraphrases, and gives a review of paraphrasing techniques.

[3] http://www.lemurproject.org/clueweb09/index.php

As we have seen, none of the discussed researches presents a dataset for paraphrased plagiarism detection. The standard solution (TIRA) for PDS evaluation does not have a dataset for Russian. There is no study for Russian covering techniques that unscrupulous authors use (can use) during the writing of plagiarized texts to hide the fact of plagiarism. The current paper will study all these aspects.

3. Creating the dataset

3.1. Common considerations

The creation of the ParaPlag was inspired by our own need to evaluate quality of our PDS which implements some original plagiarism detection methods for English and Russian. The PAN CLEF provides a great opportunity to test them but has some significant limitations. Participants need to use two standard search engines, and tasks do not contain texts in Russian. An alternative approach to evaluate the quality of plagiarism detection for texts in Russian is to use the available results of plagiarism investigations done by experts based on Russian Ph.D. theses repository—Dissernet[4]. However, in most cases each document from Dissernet contains text reuse from a few sources as a pure "copy-paste" with minimal changes. So it does not contain any significant paraphrase (or it was not be marked by experts as text reuse).

At the early stage of our research, we have considered approaches for automated generation of paraphrased plagiarized data. However, the automatic paraphrase (even synonymization) of the given text is a quite challenging task if we want to keep the original sense. The synonymization tools are widespread but their automatic usage makes text meaningless and ugly. Therefore, we asked some students to be writers of plagiarized texts. They were motivated to produce non-original texts and hide plagiarism whenever possible. However, our writers are not professional plagiarists in general. They are not also experts in linguistics or information retrieval. Therefore, we provide a guide describing the writing process and set up general requirements.

The results of writings are "essays" on different topics chosen on authors wish and interest. We tried to avoid the duplication of topics so we maintain a registry of topics for essays. By doing this we address the sources duplication problem, which will be discussed later.

Essays were written in Russian using special format (Microsoft Excel sheets), so we can extract a markup and transform it into different tasks related to a PDS evaluation. The file with an essay contains the following fields: number of fragment, filename of source document (empty for original fragments), rewritten fragment (the text of an essay), source fragments (taken from source document), and applied rewrite techniques.

Writers are free to find sources for their topics on the Web and use documents in common formats (plain text, HTML, PDF, Microsoft Word).

According to the guide, our writers should work at the sentence level, so atomic text fragment, which could be reused, is one sentence. The motivation is that each sentence expresses a statement, question, exclamation, request, command or suggestion,

[4] https://www.dissernet.org/

which could be taken from source text and paraphrased. In general, modern PDS perform well in case when unscrupulous authors change the sentence order in non-original text. Therefore, we did not introduce special requirements on sentence ordering. Writers can mix sentences from different sources and sometimes insert original sentences between plagiarized sentences.

To summarize, essays contain original and paraphrased fragments, which are produced by writers with the following rewrite techniques.

3.2. Text rewrite techniques

We consider the following most common techniques, which are often used by authors to modify the original sentences and hide reused text from PDS:

1. DEL—Delete some words (about 20%) of the original sentence;
2. ADD—Add some words (about 20%) into the original sentence;
3. LPR (Light Paraphrase)—for **Essays-1:** Replace some words or phrases of the original sentence with synonyms, reorder clauses, add new words. For **Essays-2:** change word forms (number, case, form and verb tense, etc.) for some words (about 30%) in the original sentence;
4. SHF (shuffling)—Change the order of words or clauses in the original sentence;
5. CCT (concatenation)—Concatenate two or more original sentences into one sentence;
6. SEP or SSP (sentence splitting/separation)—Split the original sentence into two or more sentences (possibly with a change in the order they appear in the text).
7. SYN (synonymizing)—Replace some words or phrases of the original sentence with synonyms (e.g. "sodium chloride"—"salt"), replace abbreviations to their full transcripts, and vice versa, replace the person's name with the name initial, etc.
8. HPR (Heavy Paraphrase)—Complex rewrite of the original sentence, which combines 3–5 or even more aforementioned techniques. This type involves significant changes of the source text by paraphrase using idioms, synonyms for complex structures, a permutation of words or parts of a complex sentence, etc. Usage of this technique produces strongly paraphrased texts. So in some cases even the experts hardly to consider the rewritten text as plagiarized.
9. CPY—Copy the sentence from source and paste it into essay almost with no changes.

3.3. Writing the essays

We have prepared two tasks for our writers. The rules for the first task (**Essays-1**) were the following:

a) Each essay must contain at least *150* sentences (sentences shorter than 3 words are not taken into account);
b) Plagiarized sentences should be taken from at least 5 different source documents;
c) Each sentence must be ether rewritten from source using one of aforementioned techniques (*1–3, 5–6, 8–9*) or must be original. An author should specify an applied technique for each sentence;

d) The ratio of sentences with techniques used to rewrite them and amount of original fragments was limited. The soft limits were set as follows: *original sentences ~10–40%., CPY ~5–30%, DEL ~20–30%, ADD ~15–25%, ~LPR~10–30%, CCT ~5–15%, SEP ~5–15%, HPR ~5–20%*. However, these limits can vary from writer to writer;

e) Techniques 5–6 allow light modification of sentence (addition /deletion of 10–15% of words).

After collecting some data for **Essays-1** task, we have changed the rules and formed the second task (**Essays-2**):

a) Each essay shall contain at least *100* sentences (sentences shorter than 3 words are not taken into account), and *at least 150 sentences from sources should be used*;

b) Plagiarized sentences should be taken from at least 5 different source documents;

c) Each sentence either must be rewritten from source using *several* (more than one!) aforementioned techniques (*1–8*) or must be original. *"Copy-and paste" text reuse is not allowed*. An author should specify all applied techniques for each sentence;

d) The ratio of sentences with techniques used to rewrite them and amount of original fragments was limited as follows: *original sentences ~5–10%., CPY ~0%, HPR ~20–40%. Other technique at least 10% for each type.* However, these limits can vary from writer to writer. *If some fragment was strongly changed, so one cannot clearly define the applied techniques, it is possible to mark this fragment with the HPR type. In other cases, all techniques must mark the considered fragment.*

There is the additional limitation for the both tasks: each writer shall prepare no more than 10 essays.

Using the two tasks for our writers, we have collected the two testing subsets: **Essays-1** and **Essays-2**, which have different characteristics. **Essays-1** contains mostly essays with large amount of "atomic" usage of the paraphrasing techniques. **Essays-2** is a little bit more complex test set with large amount of heavily paraphrased fragments.

We have had very responsible writers but always remember the principle "errare humanum est". Therefore, we developed validating tools to ensure writers understand and fulfill our requirements. Tools automate detection of common errors, so supervision process gets simple. Tools control some characteristics of written essays such as percentage of techniques used, misspells in names of techniques. Tools check that sentences rewritten with DEL have less word, and sentences rewritten with ADD have more words than original. For LPR-sentences, tools control the grammatical form changing. Tools also ensure that source sentences can be found in corresponding source documents, and original sentences are not taken from this sources. Thus, all essays written under the first and second tasks are validated with a help of these tools, and writers usually correct found errors.

Both subsets will be yet another dataset for text alignment with paraphrased plagiarism. To set up a complex task for source retrieval we must hide source documents in large dataset and deal with some issues with it.

3.4. Building the background dataset for source retrieval

Building the background set of documents comprises the two preliminary steps: web crawling and plain text extraction. Both steps were done using Exactus Expert crawling subsystem (Osipov, G., et al., 2016). Documents were crawled from the Web sources: Russian Wikipedia[5], Cyberleninka[6] and Student Essays[7]. We have added sources from written essays to it. After that, a plain text was extracted from all documents and a unique numeric id was assigned to each document. We also provide a mapping from essays to sources into the dataset for training PDS on source retrieval task.

In fact a simple combining a large dataset of documents from the Web and sources from essays can give biased results in source retrieval competition, since there could be (and actually there are) a lot of near-duplicates. Near-duplicates share almost identical content, so if there are near duplicates for some sources of essays, they likely will be found by competitors. However, these findings will be treated as false positives, since they are not in original mapping that comes with essays. PAN source retrieval track deals with this problem using near-duplicate detection. The same problem appears even if source and some other document are not near-duplicates but share some text fragments. Obviously, this could affect the results of PDS evaluation.

We decided to address this problem on the stage of building of our dataset. We have indexed all crawled documents using TextApp: the search and analytical engine—the successor of Exactus Expert (Osipov, G., et al., 2016). After that, we filtered out all near-duplicates to sources, which came from essays. We use the function of TextApp[8] that searches for topically similar documents for a given query document (Suvorov, R. E., Sochenkov, I. V., 2015). It is rather similar to the inverted index based approaches (Ilyinski, S., et al., 2002), (Ageev, M. S., Dobrov, B. V., 2011), but uses not only single words but also noun phrases as features to represent documents. Thus, we are ready to present the first version of ParaPlag: the Russian dataset for paraphrased plagiarism detection.

3.5. The dataset statistics

As of writing, our volunteers continue to work on additional essays of type 2 (**Essays-2**), which will be suitable for future PDS training and testing. However, we are ready to present the current statistics on our dataset (table 1).

The subset **Essays-1** contains 118 documents, whilst the subset **Essays-2** contains 34 documents currently.

Table 2 presents the statistics on distribution of text rewrite techniques used by writers in **Essays-1**. As we have said before, in this subset each fragment is marked with the technique used to rewrite it.

[5] https://ru.wikipedia.org

[6] http://cyberleninka.ru

[7] http://studopedia.ru, http://www.bestreferat.ru, http://allbest.ru, http://do.gendocs.ru,

[8] http://demo.textapp.ru/

Table 1. ParaPlag documents statistics[9]

Source	Documents count	Comments
Cyberleninka	1,037,540	Crawled on August, 2016
Russian Wikipedia	1,330,783	Used the official dump on August, 2016[9]
Student Referats	3,325,255	Crawled on November, 2016
Academic texts	12,183	
Sources from essays	2,037	
TOTAL:	**5,707,798**	

Table 2. Distribution of text rewrite techniques in **Essays-1**

Technique	Fragments count
CPY:	1,596
LPR:	2,870
HPR:	1,839
ORIG:	1,956
DEL:	3,970
ADD:	2,930
CCT:	1,198
SSP:	1,627
TOTAL:	**17,986**

Table 3. Distribution of text rewrite techniques in **Essays-2**

Technique	Fragments count
LPR:	993
HPR:	938
ORIG:	274
DEL:	1,450
ADD:	1,231
CCT:	490
SSP:	29
SHF:	750
SEP:	366
SYN:	1,508
TOTAL:	**8,029**

Table 4. Most popular combinations of text rewrite techniques in **Essays-2** and their distribution[10]

Techniques	Fragments count
DEL, SYN:	709
ADD, SYN:	669
DEL, ADD:	625
LPR, SYN:	518
LPR, DEL:	487
SHF, SYN:	409
LPR, ADD:	400
DEL, SHF:	359
DEL, ADD, SYN:	327
ADD, SHF:	315
HPR, SYN:	303
HPR, ADD:	296

Tables 3 and 4 show the distribution of text rewrite techniques and their combinations used by writers in **Essays-2**. Each fragment of essay in this subset is marked by at least two techniques used to paraphrase it.

[9] https://dumps.wikimedia.org/

[10] The top 12 frequent combinations, which appear at least 99 times

We performed comparison of plagiarized sentences with sentences from sources. Like in (Potthast et al., 2010) we used N-gram vector space model (VSM) where N ranges from 1 to 8 words. We performed following preparations: words were normalized (via pymorphy2 (Korobov, 2015)), stop words were removed (prepositions, conjunctions, participles), N-grams were TF-weighted. The cosine measure was employed to compute similarity between sentences. Figures 1 and 3 show the obtained similarities for **Essays-1** and **Essays-2** respectively. The box plots show the middle 50% of the respective similarity distributions as well as median similarities. The high value of similarity under 1-gram VSM indicates that essays and sources are about the same topic, since they share considerable amount of their vocabulary. The varying decrease of similarity under N-gram VSM (N>2) pinpoints the difference between two collections.

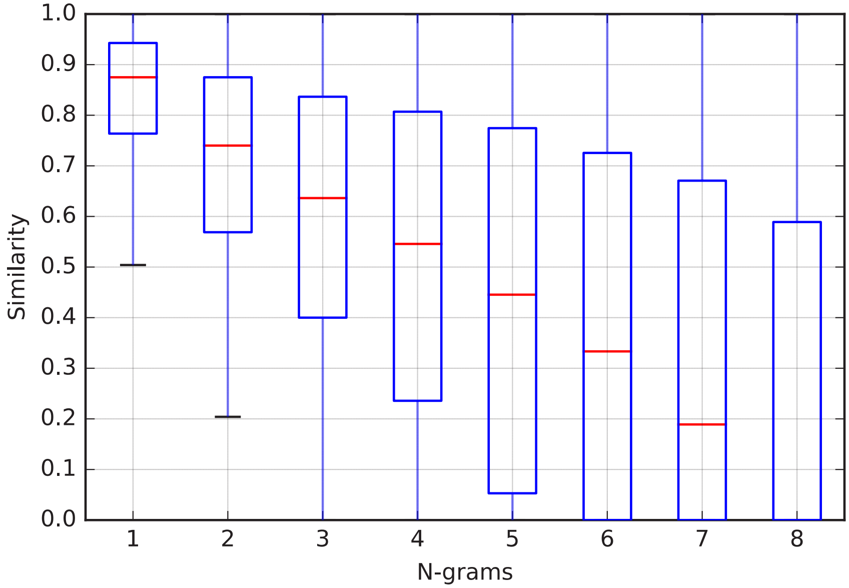


Fig. 1. Distribution of measured similarities for **Essays-1**

Each box plot shows the middle range of the distribution of measured similarities. The top of each box is the 75th percentile, the bottom is the 25th percentile, and the line in a box is a median of distribution. The upper and lower caps show 95th and 5th percentiles respectively.

Essays-1 collection contains copy-paste fragments, therefore its average similarity is relatively high even for large N (such as 6, 7). It means that essays from this collection can be found quite easily with the common methods (so-called shingles) and do not pose serious difficulties for participants of the source retrieval task. However, the collection contains disguised plagiarized text—64% of all sentences, among them: 38% with the light obfuscation techniques (ADD, DEL) and 26% with moderate or heavy obfuscations (LPR, HPR). It makes this collection appropriate for text alignment task. The measured similarity for each obfuscation type is presented in the figure 2.

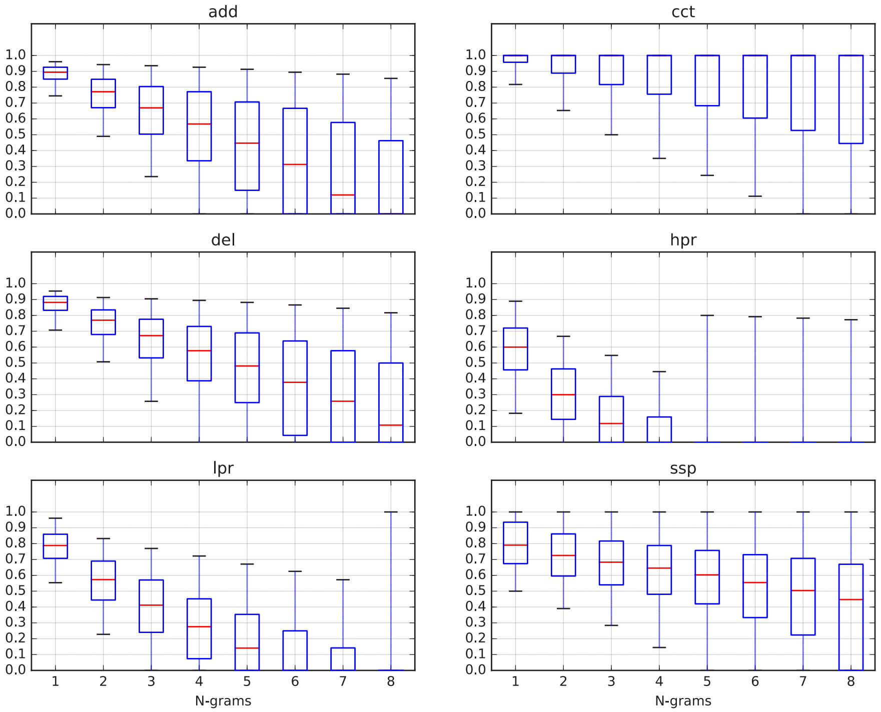


Fig. 2. Distribution of measured similarities per obfuscation type for **Essays-1**

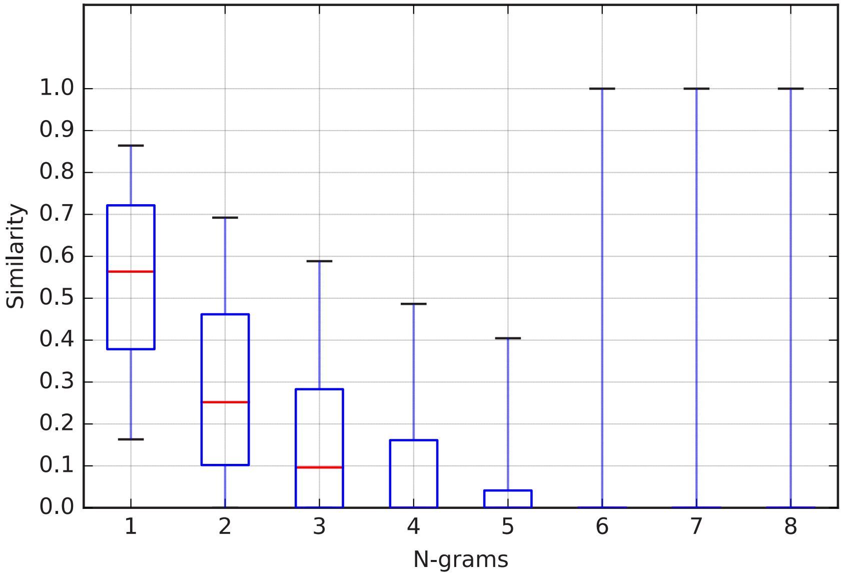


Fig. 3. Distribution of measured similarities for **Essays-2**

This figure emphasizes the aforementioned characteristics of obfuscation techniques: as DEL, ADD being easy for detection, LPR, HPR being moderately or heavily paraphrased text and CCT, SSP being almost verbatim compilation/decompilation of a source text. CCT and SSP were meant to introduce obfuscation via destruction of the structure of reused sentences and there were no special requirements for disguising of a text.

The decrease of similarity for **Essays-2** is quite steep. The main difference from **Essays-1** is the lack of any copy-paste text from the sources. There are 30–60% of 3-gram in common only for 25% of all sentence pairs. It means that source retrieval and text alignment performed on this collection can be a challenging task. Figure 4 shows the distribution of similarity for each obfuscation type of **Essays-2** collection.

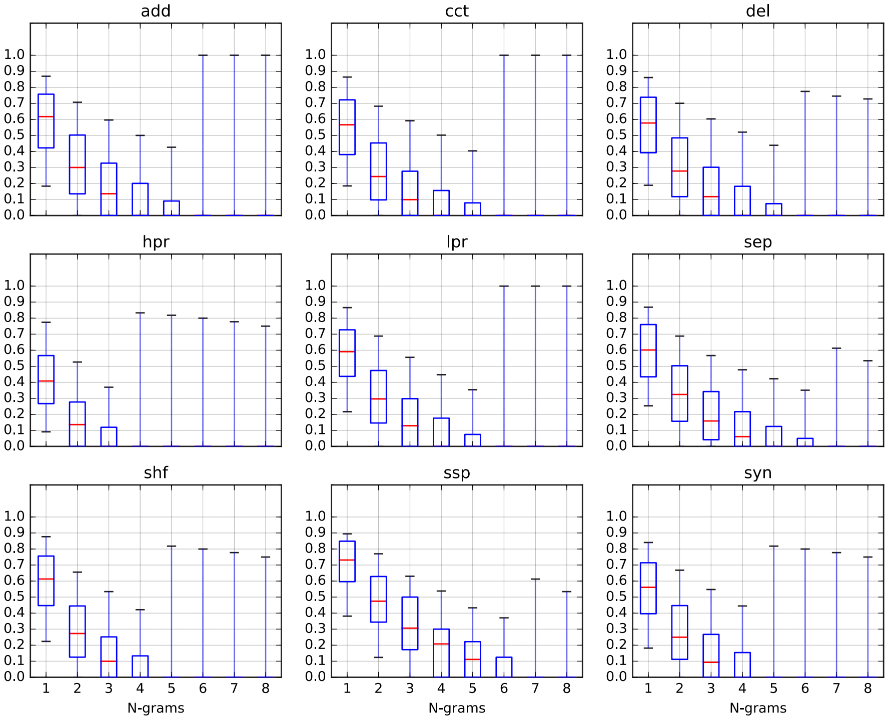


Fig. 4. Distribution of measured similarities per obfuscation type for **Essays-2**

There is a similar pattern for all distributions, which reflects the distribution of similarity for all pairs. It can be explained that these techniques were used usually together, not separately as was the case in the collection **Essays-1**. Another difference from **Essays-1** is the usage of CCT technique. It commonly indicates that a passage of a text (3–6 sentences) were used to produce short summary.

4. ParaPlag as a training dataset on PlagEvalRus-2017

The Russian Plagiarism Evaluation Seminar uses the ParaPlag as a primary dataset. The organizing committee decided to use **Essays-1** and **Essays-2** subsets as a training data for source retrieval and text alignment tracks. They also have automatically generated copy-paste and paraphrased essays to evaluate quality metrics on a big test set. The independent essay writers were encouraged to prepare additional test set similar to **Essays-2**. For this test set writers use TextApp as a search engine with indexed ParaPlag to find sources for their topics. Therefore, they do not extend the sources set. Three testing subsets (manually written essays, generated copy-paste and generated paraphrased essays) were merged and offered to competitors as a tasks in the form of plain text. Competitors do not know the mappings to sources and alignment for this training data. Thus, they should send the results of their PDS on these tasks. Finally, the organizers will calculate quality metrics according the source retrieval and text alignment tracks.

5. Conclusion and future work

We presented the ParaPlag: the Russian dataset for evaluating methods for paraphrased plagiarism detection. The ParaPlag is open and available on the Web[11]. It is used as one of the main datasets on PlagEvalRus-2017 competition. We plan to analyze the participants feedback and provide the updated version of this dataset. Hope it helps to advance the quality of modern PDS. We will continue our work on the typology of techniques used to paraphrase text and hide the plagiarism.

We plan to develop an integrated plagiarism detection task that encourages competitors to solve both source retrieval and text alignment in one track using their own plagiarism retrieval engines. The idea is that competitors need to find sources first and then to align plagiarized fragments, so these two stages could not be optimized separately.

In addition, the ParaPlag can be developed in other directions. As we have previously said, unscrupulous authors can use tools and methods to prevent correct text extraction from plagiarized documents in common formats. It is possible to investigate the ways that such tools "bring noise" to the documents to disturb text extraction procedures. Developing the test dataset of such obfuscated documents in different formats can boost methods that can detect and withstand "noise bringing" tools. Another challenging task for the future is to create a parallel (i.e. English–Russian) dataset for translated plagiarism detection.

Acknowledgments

We are grateful to students of Peoples' Friendship University of Russia (RUDN University), and Higher School of Economics (HSE) for writing essays. Our special thanks to Ph.D. student Margarita Kamenskaya for management of that process. The research is supported by RFBR, project № 16-37-60048 (mol_ad_dk).

[11] https://plagevalrus.github.io/content/corpora/paraplag.html

References

1. *Ageev, M. S., Dobrov, B. V.* (2011), An efficient nearest neighbours search algorithm for full-text documents. Vestnik S.-Petersburg Univ. Ser. 10. Prikl. Mat. Inform. Prots. Upr., (3), pp. 72–84.
2. *Agirre, E., Baneab, C., Cardiec, C., Cerd, D., Diabe, M., Gonzalez-Agirrea, A., & Mihalceab, R.* (2015), Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp. 252–263.
3. *Burrows, S., Potthast, M., & Stein, B.* (2013), Paraphrase acquisition via crowdsourcing and machine learning. ACM Transactions on Intelligent Systems and Technology (TIST), 4(3), p. 43.
4. *Clough, P., Gaizauskas, R., Piao, S. S., & Wilks, Y.* (2002), METER: Measuring text reuse. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 152–159.
5. *Dolan, W. B., Brockett, C.* (2005), Automatically constructing a corpus of sentential paraphrases. In of The Third International Workshop on Paraphrasing. M. Dras and K. Yamamoto, Eds. Kazuhide Yamamoto, Jeju, South Korea, pp. 1–8.
6. *Gollub, T., Stein, B., Burrows, S. and Hoppe, D.,* (2012), TIRA: Configuring, executing, and disseminating information retrieval experiments. In Database and expert systems applications (DEXA), 2012, pp. 151–155.
7. *Korobov M.* (2015), Morphological Analyzer and Generator for Russian and Ukrainian Languages, Analysis of Images, Social Networks and Texts, pp 320–332.
8. *Madnani, N., Dorr, B. J.* (2010), Generating phrasal and sentential paraphrases: A survey of data-driven methods. Computational Linguistics, 36(3), pp. 341–387.
9. *Osipov, G., Smirnov, I., Tikhomirov, I., Sochenkov, I., Shelmanov, A.* (2016), Exactus Expert—Search and Analytical Engine for Research and Development Support. In Novel Applications of Intelligent Systems. Springer International Publishing, pp. 269–285.
10. *Ilyinski, S., Kuzmin, M., Melkov, A. & Segalovich, I.* (2002), An efficient method to detect duplicates of web documents with the use of inverted index, in "Proceedings of 11th International Conference on World Wide Web", Honolulu, Hawaii,
11. *Potthast M., Stein B., Barrón-Cedeño A., Rosso P.* (2010), An evaluation framework for plagiarism detection, Proceedings of the 23rd international conference on computational linguistics: Posters, Beijing, pp. 997–1005.
12. *Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., & Welsch, C.* (2012), ChatNoir: a search engine for the ClueWeb09 corpus. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp. 1004–1004.
13. *Potthast M., Hagen M., Völske M., Stein B.* (2013), Crowdsourcing Interaction Logs to Understand Text Reuse from the Web, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 13) pp. 1212–1221
14. *Pronoza, E., Yagunova, E., & Pronoza, A.* (2016). Construction of a Russian paraphrase corpus: unsupervised paraphrase extraction. In Information Retrieval. Springer International Publishing, pp. 146–157.
15. *Suvorov, R. E., Sochenkov, I. V.* (2015), Establishing the similarity of scientific and technical documents based on thematic significance. Scientific and Technical Information Processing, 42(5), pp. 321–327.

MORPHORUEVAL-2017: AN EVALUATION TRACK FOR THE AUTOMATIC MORPHOLOGICAL ANALYSIS METHODS FOR RUSSIAN

Sorokin A. (alexey.sorokin@list.ru)[1,2,6],
Shavrina T. (rybolos@gmail.com)[4,6],
Lyashevskaya O. (olesar@yandex.ru)[4,8],
Bocharov V. (victor.bocharov@gmail.com)[3,5],
Alexeeva S. (sv.bichineva@gmail.com)[3],
Droganova K. (kira.droganova@gmail.com)[4,9],
Fenogenova A. (alenka_s_ph@mail.ru)[4,7],
Granovsky D. (dima.granovsky@gmail.com)[3]

[1]Lomonosov Moscow State University, [2]MIPT, [3]OpenCorpora.org, [4]National Research University Higher School of Economics, [5]Yandex, [6]GICR, [7]RDI KVANT, [8]Vinogradov Institute of the Russian Language RAS, [9]Charles University

MorphoRuEval-2017 is an evaluation campaign designed to stimulate the development of the automatic morphological processing technologies for Russian, both for normative texts (news, fiction, nonfiction) and those of less formal nature (blogs and other social media). This article compares the methods participants used to solve the task of morphological analysis. It also discusses the problem of unification of various existing training collections for Russian language.

Key words: shared task, morphological tagging, morphological parsing, parsers for Russian, universal dependencies, automatic morphological analysis, POS tagging, disambiguation, taggers

MORPHORUEVAL-2017: ОЦЕНКА МЕТОДОВ АВТОМАТИЧЕСКОГО МОРФОЛОГИЧЕСКОГО АНАЛИЗА НА МАТЕРИАЛЕ РУССКОГО ЯЗЫКА

Сорокин А. (alexey.sorokin@list.ru)[1,2,6],
Шаврина Т. (rybolos@gmail.com)[4,6],
Ляшевская О. (olesar@yandex.ru)[4,8],
Бочаров В. (victor.bocharov@gmail.com)[3,5],
Алексеева С. (sv.bichineva@gmail.com)[3],
Дроганова К. (kira.droganova@gmail.com)[4,9],
Феногенова А. (alenka_s_ph@mail.ru)[4,7],
Грановский Д. (dima.granovsky@gmail.com)[3]

[1]МГУ им. М.В.Ломоносова, [2]МФТИ, [3]OpenCorpora.org, [4]Национальный Исследовательский университет «Высшая школа экономики», [5]ООО «Яндекс», [6]ГИКРЯ, [7]НИИ КВАНТ, [8]Институт русского языка им. В. В. Виноградова РАН, [9]Карлов Университет (Прага)

MorphoRuEval-2017 — соревнование по морфологической разметке, призванное стимулировать развитие технологий морфологической обработки текстов на русском языке, в особенности текстов из сети Интернет, как нормативных (новости, литературные тексты), так и менее формального характера (блоги и другие социальные медиа). Данная статья посвящена сравнению методов, использованных командами-участниками соревнования, а также проблемам унификации различных существующих обучающих коллекций для русского языка.

Ключевые слова: соревнование по морфологическому анализу, частеречная разметка, автоматическая морфологическая разметка, алгоритмы морфологической разметки для русского языка, снятие омонимии

1. Introduction

Russian morphology has a long history of extensive research, both theoretical and practical. While theoretical science faces a wide range of problems concerning distinction of parts of speech and classification of grammatical categories [Sichinava 2011], practice of NLP usually finds temporary solutions, which occur less or more acceptable and convenient. There are already several morphological tagsets for Russian, all of them derived from different approaches such as MSD for Russian, AOT tags, OpenCorpora.org tags, Russian Positional Tagset, Natural Language Compilator tagset, etc. Generally, these tagsets are not convertible into each other without loss of information. There already exist several solutions dealing with this problem[1], yet there is no single candidate to use as reference tagset, e. g. in evaluation tracks. The only open and internationally acknowledged tagset—MSD—is overly fine-grained for the purpose of tagset unification. The morphological data standard for shared task should be 1) concise 2) compatible with international shared task results 3) suitable for rapid and consistent annotation by a human annotator 4) suitable for computer parsing with high accuracy 5) easily comprehended and used by a non-linguist (last 3—Manning Laws [Nivre, 2016])). A plausible solution of the problem is a new standard of multilingual morphological and syntactic tagging—Universal Dependencies[2] (UD) [Nivre et al. 2016]. UD initiative has developed 70 treebanks for 50 languages with cross-linguistically consistent annotation and recoverability of the original raw texts, and apparently the UD standard is becoming the main annotation paradigm for many languages. Continuing the tradition of independent evaluation of the methods used in Russian language resources and linguistic tools [Lyashevskaya et al. 2010; Toldova et al. 2012], we designed this evaluation track of Russian morphological analysis methods in order to inspire the development of the morphological taggers. For that purpose we presented the original training set which was annotated in a single format consistent with UD guidelines.

2. Evaluation tracks

Within the competition framework, we relied heavily on the experience of the previous morphological forum of Dialogue Evaluation [Lyashevskaya et al. 2010]. However, we decided to refuse organizing the track without disambiguation: participants should give only one answer for each token even if it requires disambiguation, which is the question of interest in our case. Another innovation in such campaigns for Russian is dividing the competition on the basis of model training conditions: an open track and a closed one.

1. **Closed track:** the participants are allowed to train their models only on provided data. Mostly, it is convenient for research groups and student teams who do not have large data collections. To verify the results, participants of this track are required to make their code publicly available on github, both for organizers and other participating teams. This track was intended for comparison of various tagging algorithms. Since no dictionary in competition format is available, the participants of the closed track might use their own dictionaries as well after converting them to competition format.

2. **Open track:** track members are allowed to bring any data for learning (this regulation is more appropriate for enterprise participants presenting their products).

For both tracks we provide the following evaluation (see 5):

- POS-tagging;
- tagging of the categories of interest;
- lemmatization.

[1] https://github.com/kmike/russian-tagsets

[2] http://universaldependencies.org/

The key goal of the competition was to test comparative strength of different tagging methods in two setups: a closed one, which evaluates the ability of the algorithm to learn from limited data, and the open, which allows the tagger to use any possible source of data. Since the 90-s, the state-of-the-art in morphological tagging were variants of Hidden Markov Models, where the probability of the next tag was calculated either using ngram models, as in TnT Tagger [Brants, 2000] or by the means of decision trees, as in Tree Tagger [Schmid, 1995]. For English they were beaten by conditional random fields [Sha, Pereira, 2002] and dependency networks [Toutanova, 2002], however, for the languages with developed inflected morphology their advantage is not so clear, if any, since the number of features grows too fast with the order of model. Therefore when using CRF for tagging, for example, Czech or German, one has to make decoding more complicated [Muller, 2013]. Recent advances of neural networks in POS tagging [Huang, 2015] makes them a perspective candidate.

There is no clear benchmark for morphological tagging for Russian. The previous competition organizers [Lyashevskaya et al. 2010] give no analysis of results; a recent work of [Dereza et al., 2016] shows that HMM-based approach combined with decision trees realized in Tree Tagger are substantially ahead of others, however they give no error analysis and their results are not reproducible. Two main features of Russian are free word order and regular homonymy between different forms of the same word (e. g., nominative and accusative of inanimate noun) which cannot be resolved by immediate context of the word. Hence the applicability of standard HMM or CRF approaches is limited since they cannot capture, for example, the coordination between the noun and the verb in the sentence in case these words are divided by more than 2 words. Therefore it is not clear, whether the usage of more powerful methods of machine learning or more linguistically-oriented algorithms is more beneficial. One of the goals of current competition was to investigate this dichotomy.

3. Participants

The competition was joined 15 research groups from 7 universities and research institutes (MSU, NSU, MIPT, NRU HSE, ISPRAS, NRCKI, MIEM) and 5 companies (Abbyy, OnPositive, Pullenti, Samsung R&D Institute Moscow, IQMEN) and also 3 independent researchers.

The competition resulted in 11 teams providing their materials for the closed track, and 5 teams for the open one. 1 participant have succeeded to take part in both tracks (with slight improvement on open track). About half of the teams have presented their results with lemmatization, while 7 have provided only their tagging.

4. Collecting the training data

It was decided to collect as much as possible of the annotated data in a single format for training, and additionally, to provide a sufficient number of plain texts of different genres, for participants to obtain lexical frequencies, information on compatibility and syntactic behavior, vector embeddings, etc. In total, MorphoRuEval-2017 provided the following resources:

plain texts:

1) LiveJournal (from GICR) 30 million words
2) Facebook, Twitter, VKontakte—30 million words[3]
3) Librusec—300 million words

annotated data:

1) RNC Open: a manually disambiguated subcorpus of the Russian National Corpus—1.2 million words (fiction, news, nonfiction, spoken, blog)
2) GICR corpus with the resolved homonymy—1 million words
3) OpenCorpora.org data—400 thousand tokens
4) UD SynTagRus—900 thousand tokens (fiction, news)

To unify the representation of the marked data, the conll-u format was chosen, as the most common and convenient, and for the unification of morphological tags—the format of the Universal Dependencies (further UD) 2.0 (with some specifications, see below). Resulting text collections are now available under CC BY-NC-SA 3.0 license.

4.1. Unification of the morphological tagset in annotated data

Remaining within the UD framework, we nevertheless decided to abandon some of the agreements adopted in this format to facilitate the procedure for unifying the training set. As part of the unification, we did not set the task of reducing the whole tokenization to a single variant, and we specified some complex tokens existing in GICR and UD Syntagrus data (they received the label “H”).

We omitted two POS tags SYM (symbol) and AUX (auxiliary verb), keeping in out collection the following part-of-speech categories: noun (NOUN), proper name (PROPN), adjective (ADJ), pronoun (PRON) numeral (NUM), verb (including auxiliary, VERB), adverb (ADV), determinant (DET), conjunction (CONJ), preposition (ADP), particle (PART), interjection (INTJ). Also on the data are marked punctuation marks (PUNCT) and non-word tokens (X).

The following categories are annotated:

1. Noun: gender, number, case, animate
2. Proper name: gender, number, case
3. Adjective: gender, number, case, brevity of form, degree of comparison
4. Pronoun: gender, number, case, person
5. Numeral: gender, case, graphic form
6. Verb: mood, person, tense, number, gender
7. Adverb: degree of comparison
8. Determinant: gender, number, case
9. Conjunction, preposition, particle, parenthesis, interjection, other: none

[3] We have collected posts and comments from random users and political posts for recent 5 years, fuzzy deduplication has been done to decrease the effect of popular and spam messages.

Table 1. Annotated categories for different parts of speech

Case	nominative—Nom, genitive—Gen, dative—Dat, accusative—Acc, locative—Loc, instrumental—Ins
Gender	masculine—Masc, feminine—Fem, neuter—Neut
Number	singular—Sing, plural—Plur
Animacy	animate—Anim, inanimate—Inan
Tense	past—Past, present or future—Notpast
Person	first—1, second—2, third—3
VerbForm	infinitive—Inf, finite—Fin, gerund—Conv
Mood	indicative—Ind, imperative—Imp
Variant	short form—Brev (no mark for complete form)
Degree	positive or superlative—Pos, comparable—Cmp
NumForm	numeric token—Digit (if the token is written in alphabetic form, no mark is placed).

In order to increase the annotation agreement in the collections converted from different sources, the following decisions were made (most of them follow the guidelines of UD SynTagRus corpus):

1) DET is a closed class which includes 30 pronouns used primarily in the attributive position.
2) Predicative words. Modal words such as *можно* ‘can’, *нельзя* ‘cannot’ are considered as adverbs. The word *нет* ‘no, not’ is considered as verb. The predicative words homonymous to the short neuter forms of adjectives are coded as adjectives. Therefore, short adjectives always form a part of the predicate, while adverbs do not, which can be checked semi-automatically at least in sufficient fraction of cases. This solution was accepted to facilitate automatic verification and unification of different annotated corpora since they follow different disambiguation standards and even these standards often are not realized consistently. Moreover, even in the simplest cases the border between different categories is rather vague. Our final solution coincides with UD SynTagRus guidelines after joining together short adjectives and predicatives.
3) The lemma of the verb is its infinitive form in a particular aspect (perfective or imperfective). The gerund forms constitute a part of the verb paradigm. Since the voice category was excluded, verbs ending with reflexive verb suffix *–ся* also had *–ся* in their infinitive form (the infinitive of *пишется* in *книга пишется писателем* is *писаться*, not *писать*).
4) The participles are treated as adjectives and their lemma is the Nominative masculine singular form. This was done to avoid border cases between adjectives and participles. Therefore voice category is irrelevant both for participles and other types of verbs and it was excluded from competition evaluation
5) The ordinal numerals are considered as adjectives.
6) The tense forms of the verb are divided into Past and Notpast (present or future). Aspect is not evaluated to avoid problems with biaspectual verbs.

7) The analytic (multi-word) forms of verbs, adjectives, and adverbs are not coded. For example, the analytic future tense form is annotated as two separate tokens: the future form of the verb *быть* 'to be' and infinitive.
8) SCONJ and CONJ are embraced by a single category CONJ.

A number of categories received the status of "not rated": they may be present or not in the output of the system under evaluation:

- animacy (nouns, pronouns);
- aspect, voice, and transitivity (verbs);
- pos-tags of the prepositions, conjunctions, particles, interjections, and X (others).

Several adverbs (*как 'how'*, *пока* 'while, yet'*,* *так* 'so', *когда* 'when') homonymic to conjunctions were also not rated since their annotation was controversial in different training corpora and even inside the same training corpus.

These guidelines differ a bit from those accepted in Universal Dependencies version of SynTagRus. This was done in order to simplify verification of morphological tags and their unification across corpora. Nevertheless, some inconsistency is still present. The list below summarizes the most significant differences.

RNC Open:

1) in the CONLL-u format, an extra column is provided with additional tags for typos, non-standard inflectional forms, aspect, voice, transitivity, NameType categories, etc.

GICR (the same conventions hold for the test set of the competition)

1) PROPN is tagged as NOUN.
2) A number of multi-token parenthetics as well as some other multi-word expressions (marked as H) is preserved. These multi-token constructions could also appear in the test set.

OpenCorpora:

1) Homonymy between the comparative forms of adjectives and adverbs is always resolved as the forms of adjectives (due to the agreements in the OpenCorpora dictionary)
2) the verb aspect is tagged
3) the list of possible multitoken constructions slightly differs from UD SynTagRus and GICR.

UD SynTagRus:

1) PROPN is tagged as NOUN
2) A number of multi-token expressions (marked as H) is preserved.

Since both the test set and the part of the training set used by most of the competitiors is the GICR subcorpora, we describe in more details its annotation pipeline. Initially it was automatically processed by ABBYY Compreno parser[4] providing a high-quality automatic annotation. One of the benefits of this parser is extensive usage of semantic information which helps to resolve one of the most difficult types of homonymy

[4] https://www.abbyy.com/ru-ru/isearch/compreno/

in Russian morphology, the one between accusative and nominative cases. However, as every automatic annotation, it suffers from several problems of other type. What is even more important, annotation standards and morphological system of ABBYY Compreno differ significantly from the one of UD. For example, the system always treats *это* as a demonstrative pronoun, while in UD standard and its competition dialect it was considered as a pronoun when it serves as a subject *это было трудно* 'it was difficult' and as a determiner when it is an attribute *это решение было трудным* 'this solution was difficult'. The same problem holds for the word *все* (*все пришли вовремя* 'all came on time' vs *все мои друзья пришли вовремя* ('all my friends came on time'). These ambiguities are important for the quality of annotation since pronouns are very frequent. We checked during conversion whether a particular instance of such pronouns is a undoubtful attribute (it is followed by a noun in the same case, gender and number) or a subject (for example, it is followed by a corresponding form of auxiliary verb *быть*). Analogous constraints were applied to verify and correct annotation of adverbs (for example, a potential adverb appearing between subject and verb is an actual adverb *он легко ответил* 'he easily answered') and other frequent ambiguities.

5. Testing procedure

Competitors obtain a tokenized sample as a test set. They should assign a morphological tag and (optionally) lemma to all the words in the test sample. However, only the grammemes described in Section 4 are evaluated, the presence/absence of other categories does not affect the results of evaluation. It also does not matter, which label is assigned to the words whose parts of speech are not rated, such as conjunctions, prepositions etc.

The participants should strictly follow the requirements below:

1) POS and categories labels should be taken from https://github.com/dialogue-evaluation/morphoRuEval-2017/blob/master/morphostandard
2) The tokenization of the test set is preserved. A participant should tag all the sentences in the test sample and all the words in each sentence.
3) the unique text IDs are preserved (but ignored by tagging)

We also used the following conventions

1) Both PROPN and NOUN labels for proper nouns is correct. The same holds for SCONJ and CONJ with respect to conjunctions.
2) capitalization is not significant for lemmatization.
3) *е* and *ё* are not distinguished.

5.1. Metrics

We evaluated participants performance on three test sets of different origin, News texts (Lenta.ru), fiction (Russian Magazine Hall, magazines.russ.ru) and social networks (vk.com). For each of the segments, two metrics were calculated: the percentage of correctly parsed words and the percentage of sentences whose entire parse was correct. If the participant provided lemmas, both tagging and full (lemma+tag) accuracy were evaluated, otherwise only the tag accuracy was considered. We also calculated average metrics across all three segments. For final ranking overall sentence accuracy was used since usually a correct parse of the whole sentence.

5.2. Baseline

In the review [Dereza etc., 2016] authors evaluated several taggers on the material of 6 million Russian National Disambiguated Corpus (mainly literary texts), the highest accuracy of 96,94% on POS tags and of 92,56% on the whole tagset was achieved by TreeTagger [Schmid, 1995]. This is a HMM-based tagger, which uses a binary decision tree to estimate transition probabilities. TreeTagger is also capable to tag the unknown words using a suffix/prefix lexicon. For current shared task TreeTagger was chosen as a baseline system.

We have carried on five baseline experiments (results are presented in the Table 2):

1. On the material of GICR: 75% training set, 25% test set.
2. Trained on the data of GICR: 75% training set; tested on the data of Syntagrus 25% test set.
3. On the material of Syntagrus: 75% training set, 25% test set.
4. Trained on GICR 75% training set, Syntagrus 75% training set and 1 million RNC and Opencorpora.org dataset. Tested on GICR 25% test set.
5. Trained on GICR 75% training set, Syntagrus 75% training set and 1 million RNC and Opencorpora.org data set. Tested on Syntagrus 25% test set.

Table 2. Evaluation of baseline algorithms for different training settings

Experiment	Tags	accuracy per tag	number of correct tags	accuracy per sentence	number of correct sentences
Baseline (1)	POS tag	79.49%	136,372 from 171,550	26.25%	5,456 from 20,787
	Full tag	76.54%	131,309 from 171,550	21.14%	4,394 from 20,787
Baseline (2)	POS tag	73.46%	107,846 from 146,817	9.93%	1,244 from 12,529
	Full tag	68.44%	100,482 from 146,817	6.35%	795 from 12,529
Baseline (3)	POS tag	79.19%	116,265 from 146,817	17.02%	2,132 from 12,529
	Full tag	75.43%	110,749 from 146,817	11.87%	1,487 from 12,529
Baseline (4)	POS tag	73.89%	126,759 from 171,550	23.89%	4,967 from 20,787
	Full tag	71.15%	122,054 from 171,550	18.51%	3,848 from 20,787
Baseline (5)	POS tag	72.10%	105,854 from 146,817	14.89%	1,866 from 12,529
	Full tag	69.71%	102,346 from 14,6817	11.76%	1,473 from 12,529

5.3. Golden Standard

We have provided 3 different segments from GICR for testing, all not published before, 7000 tokens each. These are 568 sentences from VKontakte, from News (Lenta ru, 353 sentences), and from modern literature, Russian Magazine Hall (394 sentences).

These materials were tagged morphologically within the framework of GICR pipeline [Selegey et al. 2016], then converted from MSD to UD 1.4 and carefully checked automatically and manually, with paying special attention to consistency of annotation, format specifications and systematic errors of automatic tagging (case homonymy, short form adjectives and adverbs, etc.). As a side result we discovered, that no existent automatic or semi-automatic procedure guarantees the quality of morphological analysis sufficient to be a "Gold standard" for parsers testing and manual verification and correction is a necessary postprocessing step. Golden standard sentences were randomly shuffled in a tokenized set of 600–900 thousand tokens for each segment.

All participant results and scripts for their comparison with golden standard are now available online[5].

6. Team results and methods

One of the goals of current competition was to compare different approaches to morphological tagging. The clear winner of the competition is ABBYY team which participated in the open track. In the closed track slightly better than others was the team of MSU, however three other teams are less than 1% behind in terms of tag accuracy, the gap for sentence accuracy is more significant.

The top-ranked participant algorithms fall in two camps. The first utilizes the power of neural networks to uncover hidden relationships between words in the sentence. It includes the winner (ABBYY) and Sagteam and Aspect from the closed track. The second group tries to use linguistic information using more complex features. It contains MSU and IQMEN teams (top 2 on closed track).

ABBYY team uses two-layer bidirectional neural network with several additional layers as a learning method. Each word is characterized by 250-dimensional embedding and additional morphological and graphic features. The model was pre-trained on additional Wikipedia corpus and these parameters were further optimized on GICR data from the training set. Wikipedia corpus was pretagged using ABBYY Compreno parser.

MSU team used an HMM classifier as a baseline model. Then n-best hypotheses obtained from this classifier were reranked using additional high-level features, such as number of coordinated adjective-noun and determiner-noun groups, number of correctly detected sentence clauses etc. They used logistic regression as reranking algorithms, which was trained on GICR data in order to assign higher score to correct sentence parses. To increase the quality of basic classifier tags in the training corpora were enriched with transitivity information for verbs and case label for nouns.

[5] https://drive.google.com/drive/folders/0B600DBw1ZmZASDFRVkJVd0pqNXM

IQMEN teams collected a set of hypotheses for each word and learnt the best one using the features for the word under consideration as well as for the word in a window of width 7 around it. Features included morphological (e.g part-of-speech, number, case, gender etc.) and graphical (suffixes, capitalization) information both for the word itself and its neighbours. The optimal tags for the sentence were guessed from left to right in a greedy fashion, instead of the tags for the words to the right the ambigiity classes were used. Similar approach was applied by Morphobabushka team, however. they refused to use any dictionaries guessing the tags for unknown words basing on their suffixes and features of surrounding words. IQMEN applied SVM with hash kernel, while Morphobabushka implemented SVM-NB classifier.

Sag team uses convolutional neural network, taking character-level representations for individual words and using several additional layers to comprise them into the representation of the whole sentence. Their algorithm does not use any dictionary except collected from the training set. Aspect team applies similar approach but use their own dictionary together with error-processing model to deal with typos and colloquial writing.

Table 3. Results of MorphoRuEval-2017

Team name	**team ID**	**Track**	**Number of the best try**	**Accuracy by tags**	**Accuracyby sentences**	**Lemmatization, accuracy by wordforms**	**Lemmatization, accuracy by sentences**
MSU-1	C	Closed	2	93.39	65.29		
IQMEN	O	Closed	1	93.08	62.71	92.22	58.21
Sagteam	H	Closed	2	92.64	58.40	80.73	25.01
Aspect	A	Closed	2	92.57	61.01	91.81	56.49
Morphobabushka	M	Closed	2	90.07	48.10		
Pullenti Pos Tagger	G	Closed	4	89.96	47.23	89.32	45.18
	B	Closed	6	89.91	48.2		
	N	Closed	4	89.86	47.13	85.10	29.04
	K	Closed	4	89.46	48.54	88.47	44.78
	F	Closed	2	88.14	39.63	87.27	36.90
	I	Closed	2	86.05	34.62		
	L	Closed	2	71.48	6.48		
ABBYY	E	Open	3	97.11	83.68	96.91	82.13
Aspect	A	Open	4	92.38	60.90	87.66	41.12
	N	Open	5	90.88	51.77	85.91	32.57
	J	Open	1	83.51	29.69		
	D	Open	5	77.13	17.19		

Neural networks approach are the clear winner, however, several remarks should be made. ABBYY team uses an additional corpus with rich annotation to train their model, it is not clear, whether their advantage would be so clear without it. On the closed task neural network methods are slightly behind more linguistically oriented approaches based on linear classifiers with rich feature descriptions. Therefore it is reasonable to ask, if neural network approach has the same benefits when only limited amount of training data is available. Interestingly, that on the SIGMORPHON-2016

[Cotterell et al., 2016] competition on morphological reinflection the same pattern was observed: elaborated neural network approaches clearly outperformed more traditional ones which attempted to utilize more linguistically motivated features. Another reasonable question is whether algorithms of different type can be combined together to compensate their weaknesses, for example, MSU team method can take any classifier as the basic one provided it ables to generate n-best lists of hypotheses together with probability estimates.

Comparing to previous evaluation of morphological parsers for Russian language, current systems show significant improvement. Indeed, the top-ranked of the [Lyashevskaya et al., 2010] competition achieved 97% result only for POS-tagging, while the winner of current competition showed the same result for entire grammatical tags. The top-system result is comparable with results for other inflective languages with free word order and rich inflective morphology, such (95.75% for Czech in [Strakova, 2013]). Note that training corpus included only LJ posts and test corpus contained texts from three different sources, so the top-performing systems also demonstrated its ability to perform successfully not only on the domain they were trained on, but also on the texts from different origin.

7. Problems and discussion

One of the purpose of the work was to provide a unified training corpus for morphological tagging containing texts from different sources. However, it was not realized in full. As already mentioned, different corpora have different standards of lemmatization, for example for pronouns (what is the lemma of *она* 'she', *он* 'he' or *она*), but more important is that they have different standards of morphological annotation. There are many border cases treated in a different way in different corpora, such as the distinction between adverbs, predicatives and short adjectives, processing of reflexive verbs (they belong to special medial voice in GICR and RNC while UD SynTagRus distinguishes only active and passive voices) and so on. All competition participants trained their model only on GICR subset of the training corpus, which demonstrates that even after conversion to the same format joint usage of corpora of different origin and genre structure is problematic. Therefore the problem of unification is far from its solution, however, UD format looks appealing to be a destination of conversion from other formats.

For Russian language there is no dictionary in UD format, which could be used by participants and organizers to verify their decisions. In the framework of MorphoRuEval, there was carried out some work by the organizing committee, on the development of the OpenCorpora.org open-source dictionary: the dictionary was expanded by several thousand paradigms from the GICR dictionary, and then converted to universal dependencies. We hope that this dictionary will be included in the official UD documentation for Russian and will be useful in future evaluation.

8. Conclusion

Shared task on morphological tagging showed fruitful results in several important aspects:

- An original data set collected from different corpora which was annotated in a single format consistent with UD guidelines was prepared and presented;
- Comprehensive guidelines for testing procedure and evaluation were created.
- The comparison of different parsing strategies showed that neural network approach is state-of-the-art method for morphological parsing of Russian.
- dataset for future improvement of morphological parsers, comprising texts from different sources, was created.

All materials of MorphoRuEval-2017 including training and test set are now available at the competition's github[6]. We welcome NLP-researchers and specialists in machine learning to use this collection and we hope that the collection will stay practical and relevant for a long time.

Acknowledgements

Authors express sincere gratitude to all the competition participants and especially thank Mikhail Korobov and Anton Kazennikov for their help during format discussion and explicit determination of unification conditions. We thank ABBYY Compreno and GICR teams for automatic annotation of initial subset of GICR subcorpora and assistance in its conversion to competition format. We are also grateful to Ilia Karpov from Kvant, Svetlana Savchuk from RNC and members of OpenCorpora and GICR teams for providing and preparation of their resources.

The work was partially supported by the grant NSh-9091.2016.1 for leading scientific groups. The reported study is supported by RFBR grants № 15-07-09306 "Evaluation Standards in Automatic Information Retrieval for Russian" and № 16-29-09614 "Principles and technologies of machine processing for social network texts in Russian". The UD SynTagRus corpus preparation was supported by GA UK grant 794417.

References

1. *Brants T.* TnT: a statistical part-of-speech tagger. In Proceedings of the sixth conference on Applied natural language processing. — Association for Computational Linguistics, 2000. — Pp. 224–231.
2. *Cotterell Ryan, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden* (2016) The SIGMORPHON 2016 shared task—morphological reinflection. In Proc. of the 2016 Meeting of SIGMORPHON.
3. *Dereza O. V., Kayutenko D. A., Fenogenova A. S.* (2016) Automatic morphological analysis for Russian: A comparative study. In Proceedings of the International Conference Dialogue 2016. Computational linguistics and intellectual technologies. Student session (online publication). Retrieved from: http://www.dialog-21.ru/media/3473/dereza.pdf
4. *Huang Z., Xu W., Yu K.* Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991. — 2015. Retrieved from https://arxiv.org/abs/1508.01991

[6] https://github.com/dialogue-evaluation/morphoRuEval-2017

5. *Lyashevskaya, Olga, Irina Astaf'eva, Anastasia Bonch-Osmolovskaya, Anastasia Garejshina, Julia Grishina, Vadim D'jachkov, Maxim Ionov, Anna Koroleva, Maxim Kudrinsky, Anna Lityagina, Elena Luchina, Eugenia Sidorova, Svetlana Toldova, Svetlana Savchuk, and Sergej Koval'* (2010) NLP evaluation: Russian morphological parsers [Ocenka metodov avtomaticheskogo analiza teksta: morfologicheskije parsery russkogo jazyka]. In: Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue'2010. Vol. 9 (16), 2010. Pp. 318–326.
6. *Müller T., Schmid H., Schütze H.* Efficient higher-order CRFs for morphological tagging. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. — 2013. — Pp. 322–332.
7. *Nivre J.* (2016) Reflections on Universal Dependencies. Uppsala University. Department of Linguistics and Philology.
8. *Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajic J., Manning Ch. D., McDonald R., Petrov S., Pyysalo S., Silveira N., Tsarfaty R., Zeman D.* (2016), Universal Dependencies v1: A Multilingual Treebank Collection, Proc. of LREC 2016, Portorož, Slovenia, pp. 1659–1666.
9. *Schmid H.* Treetagger (1995) A language independent part-of-speech tagger. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart,1995. — Vol. 43, pp. 28.
10. *Selegey D., Shavrina T., Selegey V., Sharoff S.* (2016) Automatic morphological tagging of russian social media corpora: training and testing.In: Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue'2016.
11. *Sha F., Pereira F.* Shallow parsing with conditional random fields. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology — Association for Computational Linguistics, 2003. — Vol. 1., pp. 134–141.
12. *Sichinava D. V.* Parts of speech. [Chasti rechi. Materialy dlja proekta korpusnogo opisanija russkoj grammatiki (http://rusgram.ru)]. Moscow, ms. 2011. Available at: http://rusgram.ru/Chasti_rechi.
13. *Straková J., Straka M., Hajic J.* Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In Proceedings of ACL (System Demonstrations). — Association for Computational Linguistics, 2014. — pp. 13–18.
14. *Toldova, S., Sokolova, Elena, Astafiyeva, Irina, Gareyshina, Anastasia, Koroleva, Anna, Privoznov, Dmitry, Sidorova, Evgenia, Tupikina, Ludmila, Lyashevskaya, Olga.* Ocenka metodov avtomaticheskogo analiza teksta 2011–2012: Sintaksicheskie parsery russkogo jazyka [NLP evaluation 2011–2012: Russian syntactic parsers]. In Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue 2012. Vol. 11 (18), 2012. Moscow: RGGU, pp. 797–809.
15. *Toutanova K.. Klein D., Manning C., Singer Y.* Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. — Association for Computational Linguistics, 2003. — Pp. 173–180.

Appendix

News

team ID	track	Accuracy by tags	Accuracy by sentences	Lemmatization, accuracy by wordforms	Lemmatization, accuracy by sentences
C	closed	93.71	64.80		
O	closed	93.99	63.13	92.96	56.42
H	closed	93.35	55.03	81.60	17.04
A	closed	93.83	61.45	93.01	54.19
M	closed	90.52	44.41		
G	closed	89.73	39.66	89.04	37.71
B	closed	90.79	43.58		
N	closed	91.53	49.16	87.01	25.70
K	closed	90.36	45.53	89.23	40.22
F	closed	90.43	36.87	89.61	33.52
I	closed	88.66	29.89		
L	closed	75.88	2.790		
E	open	97.37	87.71	97.18	85.75
A	open	93.83	61.45	88.35	33.24
N	open	91.98	52.51	87.20	27.93
J	open	84.25	23.18		
D	open	79.52	10.89		

VKontakte

team ID	track	Accuracy by tags	Accuracy by sentences	Lemmatization, accuracy by wordforms	Lemmatization, accuracy by sentences
C	closed	92.29	65.85		
O	closed	92.39	64.08	91.69	61.09
H	closed	92.42	63.56	82.80	35.92
A	closed	91.49	61.44	90.97	60.21
M	closed	89.55	51.41		
G	closed	89.17	54.58	88.65	52.64
B	closed	88.96	52.29		
N	closed	88.44	48.59	83.67	34.51
K	closed	88.39	52.11	87.34	48.94
F	closed	86.72	44.72	85.81	41.90
I	closed	84.29	41.73		
L	closed	70.13	14.61		
E	open	96.52	81.34	96.26	79.93
A	open	90.92	61.09	86.97	48.24
N	open	89.63	52.29	84.58	36.80
J	open	82.87	36.44		
D	open	75.42	23.42		

Modern literature

team ID	track	Accuracy by tags	Accuracy by sentences	Lemmatization, accuracy by wordforms	Lemmatization, accuracy by sentences
C	closed	94.16	65.23		
O	closed	92.87	60.91	92.01	57.11
H	closed	92.16	56.60	77.78	22.08
A	closed	92.40	60.15	91.46	55.08
M	closed	90.13	48.48		
G	closed	90.97	47.46	90.28	45.18
B	closed	89.98	48.73		
N	closed	89.61	43.65	84.61	26.9
K	closed	89.63	47.97	88.84	45.18
F	closed	87.26	37.31	86.39	35.28
I	closed	85.21	32.23		
L	closed	68.43	2.03		
E	open	97.45	81.98	97.3	80.71
A	open	92.40	60.15	87.65	41.88
N	open	91.02	50.51	85.95	32.99
J	open	83.42	29.44		
D	open	76.45	17.26		

Algorithm description

Team	Track	Achievements	Method	additional training set (for open track only!)	Dictionary
Pullenti	closed	3rd place by mean lemmatization accuracy (by wordforms and sentences) and on VK, Modern literature	Rule-based approach, no training set used	—	Own dictionaries
Mental Computing	closed	3rd place by lemmatization (wordforms) on News	Char-level neural networks using Keras. The core algorithm is a grid classifier built using a RNN on LSTM, training on GICR data.	—	Dictionary collected from the training set
Abbyy	open	1st place by all metrics on open track	Bidirectional LSTM with probabilities and features from Abbyy NLC module, converted to UD	Pre-training on a large corpus (several tens of millions of words, including Russian Wikipedia) tagged by Compreno, then learning on GICR training data with more accurate tagging and a more suitable genre components.	Abbyy Compreno Dictionary

Team	Track	Achievements	Method	additional training set (for open track only!)	Dictionary
Sag	closed	3rd place by mean accuracy (tags and sentences) and on VK	2-layer deep learning neural network A two-level representation of a sentence by individual characters level (see Section 2.1.1) and level of words (Section 2.1.2), inspired by works [Nogueira dos Santos C., Zadrozny B.], [Zhiheng H., Wei X., Kai Y.], [Plank B., Søgaard A., Goldberg Y.]. Keras framework	—	Dictionary collected from the training set
Aspect	closed	1st place by lemmatization accuracy on news,2st place by tag accuracy on news, 3rd place by mean lemmatization accuracy (by wordforms and sentences) on all segments, closed track	Deep neural networks (based on recurrent neural networks) with the char-level representation of words	—	Own dictionaries for spell-checking and internet-slang
Aspect	open	2nd place by all metrics on open track	Deep neural networks (based on recurrent neural networks) with the char-level representation of words	Own tagged corpora of internet-texts	Own dictionaries for spell-checking and internet-slang
KZN	closed	2nd place by by mean accuracy (by wordforms and sentences) on all segments, 1st place by mean accuracy on News, 1st place by mean lemmatization accuracy on all segments	The model consists of four parts: the morphology module based on the AOT dictionary and GICR corpus, the predictive morphology module on the basis of the corpus, the SVM-classifier for removing morphological homonymy, and the context-dependent procedure for tagging the whole sentence.	—	AOT dictionary
Biser	closed	3rd place by lemmatization accuracy (by sentences) and on News, Modern literature	dictionary-based morphological guesser, homonymy is resolved using CRF.	—	
MSU-1	closed	1st place by mean accuracy (by wordforms and sentences) on all segments, closed track	Baseline HMM model. features reflecting grammatical correctness for reordering , reordering is performed using logistic regression	—	Abbyy Compreno Dictionary

LEVENSHTEIN DISTANCE AND WORD ADAPTATION SURPRISAL AS METHODS OF MEASURING MUTUAL INTELLIGIBILITY IN READING COMPREHENSION OF SLAVIC LANGUAGES

Stenger I. (ira.stenger@mx.uni-saarland.de),
Avgustinova T. (avgustinova@coli.uni-saarland.de),
Marti R. (rwmslav@mx.uni-saarland.de)

Saarland University, Saarbrücken, Germany

In this article we validate two measuring methods: Levenshtein distance and word adaptation surprisal as potential predictors of success in reading intercomprehension. We investigate to what extent orthographic distances between Russian and other East Slavic (Ukrainian, Belarusian) and South Slavic (Bulgarian, Macedonian, Serbian) languages found by means of the Levenshtein algorithm and word adaptation surprisal correlate with comprehension of unknown Slavic languages on the basis of data obtained from Russian native speakers in online free translation task experiments. We try to find an answer to the following question: Can measuring methods such as Levenshtein distance and word adaptation surprisal be considered as a good approximation of orthographic intelligibility of unknown Slavic languages using the Cyrillic script?

Keywords: Levenshtein distance, word adaptation surprisal, orthographic intelligibility, reading intercomprehension, East and South Slavic languages

РАССТОЯНИЕ ЛЕВЕНШТЕЙНА И МЕРА НЕОЖИДАННОСТИ АДАПТАЦИИ СЛОВА КАК МЕТОДЫ ИЗМЕРЕНИЯ МЕЖЪЯЗЫКОВОЙ ПОНЯТНОСТИ СЛАВЯНСКИХ ЯЗЫКОВ ПРИ ЧТЕНИИ

Штенгер И. (ira.stenger@mx.uni-saarland.de),
Августинова Т. (avgustinova@coli.uni-saarland.de),
Марти Р. (rwmslav@mx.uni-saarland.de)

Университет земли Саар, Саарбрюккен, Германия

В данной статье мы проверяем два метода оценки степени близости родственных языков — расстояние Левенштейна и меру неожиданности адаптации слова — в качестве потенциальных параметров для определения успеха межъязыкового понимания в ситуации, когда читателю необходимо извлечь информацию из текста на незнакомом языке. Мы исследуем, в какой степени орфографические дистанции между русским языком и другими восточнославянскими (украинским, белорусским) и южнославянскими языками (болгарским, македонским, сербским), установленные с помощью алгоритма Левенштейна и меры неожиданности адаптации слова, соотносятся с экспериментальными результатами понимания незнакомых славянских языков носителями русского языка. Сбор данных был выполнен на базе экспериментов в виде заданий по свободному переводу в режиме онлайн. Мы попытаемся найти ответ на следующий вопрос: могут ли такие методы измерения как расстояние Левенштейна и мера неожиданности адаптации слова оптимально оценить понятность орфографии незнакомых славянских языков, использующих кириллицу?

Ключевые слова: расстояние Левенштейна, мера неожиданности адаптации слова, понятность орфографии, взаимопонимание при чтении, восточнославянские и южнославянские языки

1. Introduction

Intercomprehension (Doyé 2005), receptive multilingualism (Braunmüller and Zeevaert 2001) or semi-communication (Haugen 1966) reveals the human ability to understand (but not speak or write) one or more unknown foreign languages that are related to at least one language in the individual's linguistic repertoire. More or less systematic mutual intelligibility investigation was undertaken for certain language groups, e.g., Scandinavian (Gooskens 2006), Germanic (Möller and Zeevaert 2015, Vanhove 2015), West and South Slavic (Golubović and Gooskens 2015). It was shown that the degree of intelligibility of an unknown but (closely) related language depends on both linguistic and extra-linguistic factors (Gooskens 2013).

In all reading activities, orthography is the primary linguistic interface for extracting information from unfamiliar encodings in the stimulus, and it critically affects the transmission of information across languages. The present study focuses on the orthographic intelligibility of written East Slavic (Ukrainian, Belarusian) and South Slavic (Bulgarian, Macedonian, Serbian) languages for Russian subjects. We consider two linguistic distance measurements as potential predictors of successful reading intercomprehension and validate them in web-based experiments.

This article is organized as follows. Section 2 gives a short overview of the Cyrillic alphabet and the main orthographic principles. Section 3 presents the online experiment of testing orthographic intelligibility and describes the experimental data on the basis of which both the normalized Levenshtein distance and the normalized word adaptation surprisal were calculated. In Section 4, the two measuring methods are correlated with the experimental results. Finally, some general conclusions are drawn and future work is outlined.

2. Cyrillic orthographic code

While an alphabet of a language consists of a set of letters (graphemes) used to compose written texts in that language (Sgall 2006), the mechanisms of orthographic code are determined by various principles underlying the established writing systems. All the languages we investigate here employ the Cyrillic alphabet, albeit with slight adaptations, i.e.[1]

Russian (RU)	а б в г д е ё[1] ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я	(33)
Ukrainian (UK)	а б в г ґ д е є ж з и і ї й к л м н о п р с т у ф х ц ч ш щ ь ю я	(33)
Belarusian (BE)	а б в г д е ё ж з і й к л м н о п р с т у ў ф х ц ч ш ы ь э ю я	(32)
Bulgarian (BG)	а б в г д е ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ь ю я	(30)
Macedonian (MK)	а б в г д ѓ е ж з ѕ и ј к л љ м н њ о п р с т ќ у ф х ц ч џ ш	(31)
Serbian (SR)	а б в г д ђ е ж з и ј к л љ м н њ о п р с т ћ у ф х ц ч џ ш	(30)

Slavic orthographies using the Cyrillic alphabet are based primarily on the phonemic principle, but they also observe other principles, e.g., phonetic, morphological, historical/etymological (Kučera 2009). For example, the Serbian orthography adheres basically to the phonemic principle, with a strong tendency towards the phonetic principle (Kučera 2009, Marti 2014). Despite the fact that Russian orthography is based in general on the phonemic principle, the morphological principle is relevant too, depending on what is understood as a phoneme (Ivanova 1991, Musatov 2012). All Slavic orthographies represent nowadays so-called mixed systems providing the respective languages with a number of general patterns (for more details see Stenger 2016).

3. Material and methods

We investigate reading intercomprehension among Slavic languages and approach the problem of their mutual intelligibility from an information-theoretic perspective in terms of surprisal, taking into consideration information en- and decoding at different linguistic levels.[2]

When research in spoken semi-communication or in reading intercomprehension focuses on testing text understanding (i.e. Beijering et al. 2008, Golubović and Gooskens 2015, Gooskens 2007), the intelligibility scores are based on the text as a whole. This means, in particular, that the influence of different linguistic factors — such as textual and sentence context, syntax, lexis, morphology, phonetics/phonology or orthography — cannot be distinguished in detail, if at all identifiable.

We wanted to determine the role of orthography in written intercomprehension, and for that reason chose to focus on isolated cognate recognition first. Even though it may

[1] The letter *ë* is generally used in dictionaries and schoolbooks only.

[2] This study was carried out within the project INCOMSLAV *Mutual Intelligibility and Surprisal in Slavic Intercomprehension*, which is part of the Collaborative Research Center 1102 *Information Density and Linguistic Encoding*.

seem artificial to test cognates without context, since the latter may provide helpful information, the underlying assumption here is that the correct cognate recognition is a precondition of success in reading intercomprehension. If the reader correctly recognizes a minimal proportion of words, he or she will be able to piece the written message together.

3.1. Experiment

The orthographic intelligibility between Russian and five other Slavic languages was tested in web-based experiments.[3] 119 native speakers of Russian between 14 and 71 years of age (average 34 years) took part in the challenge. Around three-fourths of them were female. The participants started the experiment with registration and then completed a background questionnaire in their native language. Afterwards 6 challenges were presented randomly: 2 challenges with 60 different BG stimuli in each group, 1 challenge with 60 UK stimuli, 1 challenge with 60 BE stimuli, 1 challenge with 50 MK stimuli, and 1 challenge with 50 SR stimuli. The choice of the stimuli from the manually prepared lists and the order of presentation were also randomized. The participants saw the stimuli on their screen, one by one, and were given 10 seconds to translate each word into Russian. The time limit was carefully piloted taking into consideration the experience of other experiments in reading intercomprehension. During the experiment the participants received feedback in form of emoticons for their answers. The allocated time limit seemed to be sufficient for typing even the longest words, but not long enough for using a dictionary or online translation tools. It was possible to finish before the 10 seconds were over by either clicking on the ‘Next’ button or pressing ‘Enter’ on the keyboard. After 10 seconds the participants saw the next stimulus on their screen. The results were automatically categorized as “right” or “wrong” via pattern matching with expected answers. Some stimuli had more than one possible translation. We also provided a list of so-called alternative correct answers. For example, the BE word *дзіця* (*dzicja*)[4] ‘*child*’ can be translated in Russian as *дитя* (*ditja*) or *ребёнок (rebënok*) or *ребенок (rebenok),* all meaning ‘*child*’. All these translations were counted as correct.

In the present study we exclude those participants who have indicated knowledge of the stimuli language(s) in the questionnaire and analyze the results only of the initial challenge for each participant in order to avoid any learning effects. The mean percentage of correctly translated items constitutes the intelligibility score of a given language (Table 1).

Table 1. The results of free translation task experiments

Stimuli languages	Participants’ native language: RU
UK	80.42%
BE	71.66%
BG	70.88%
MK	61.81%
SR	57.16%

[3] The web application is available at http://intercomprehension.coli.uni-saarland.de/ru/

[4] Transliteration is given according to DIN 1460.

3.2. Material

For the computational transformation experiments on parallel word sets presented in (Fischer et al. 2015), we collected and aligned parallel Slavic word lists, at first for two language pairs: Czech—Polish and Bulgarian—Russian. For each pair, a list of internationalisms and a list of Pan-Slavic vocabulary were freely available from the EuroComSlav website.[5] Additionally we compiled a third parallel list of cognates from Swadesh lists for these languages.[6] All three lists were slightly modified. Thus, formal non-cognates were removed and formal cognates, if existing, were added to the lists where the pairs in the original lists consisted of non-cognates. For example, BG–RU *ние–мы* (*nie–my*) '*we*' were removed and the BG *звяр* (*zvjar*) '*beast*' instead of *животно* (*životno*) '*animal*' was added to its RU formal cognate *зверь* (*zver'*) '*animal, beast*'. The linguistic items in these lists belong to different parts of speech, mainly nouns, adjectives, and verbs.

In the second step, we manually collected a cross-linguistic rule set of corresponding orthographical units (transforming both individual letters and letter strings) from comparative historical Slavic linguistics (e.g. Bidwell 1963, Vasmer 1973, Žuravlev et al. 1974–2012). This resulted in sets of diachronically-based orthographic correspondences, e.g. BG–RU: *б*:*бл*, *жд*:*ж*, *я*:*е*, *ла*:*оло* etc. We then tested this set of diachronically-based orthographic correspondences on the parallel word lists mentioned above. By applying the transformation rules, we categorized the cognates in the pairs as (i) identical, (ii) successfully transformed, or (iii) non-transformable by the rules. In most cases, the automatic transformations were judged to be satisfactory, e.g. BG–RU 128 correctly transformed items excluding doublets of a total of 935 items in all three lists (for more details see Fischer et al. 2015).

In addition, we carried out orthographic transformation experiments on the parallel word lists of Common Slavic vocabulary (Carlton 1991, Mel'nyčuk 1966) for the language pairs UK–RU, BE–RU, BG–RU, MK–RU, and SR–RU. The Common Slavic vocabulary consists of 212 examples for 15 Slavic languages. While the original data have some empty slots for some of the languages, the parallel vocabulary lists for the computational transformation include only 190 items for all languages, consisting mostly of nouns, with a small amount of 23 adjectives and 27 verbs in each language. The number of successfully transformed items differs in the respective pairs: 102 items for BE–RU, 76 items for UK–RU, 68 items for SR–RU, 63 items for BG–RU, and 62 items for MK–RU. The correctly transformed items from all computational transformation experiments are used as the basis for the selection of stimuli in our web-based experiments (Section 3.1). In this way we could exclude possible different derivational morphemes between related languages in order to focus on the impact of mismatched orthographic correspondences in cognate intelligibility.

[5] Pan-Slavic list: http://www.eurocomslav.de/kurs/pwslav.htm;
internationalism list: http://www.eurocomslav.de/kurs/iwslav.htm (accessed 11.07.2015).

[6] Swadesh-list: http://en.wiktionary.org/wiki/Appendix:Swadesh_lists_for_Slavic_languages.

3.3. Levenshtein distance

Orthographic distances between corresponding cognates are usually measured on the basis of the Levenshtein distance metric (Levenshtein 1966). Kessler (1995) introduced the algorithm for measuring distances between Irish Gaelic dialects. Since then it has been applied successfully not only to different dialects of one language, but also to (closely) related languages (Beijering et al. 2008, Gooskens 2006). Levenshtein distance is considered a fairly good predictor of overall intelligibility in speech semi-communication in related language varieties as well as in reading intercomprehension (Gooskens 2007, Kürschner et al. 2008, Vanhove and Berthele 2015).

The Levenshtein distance between corresponding words is based upon the minimum number of symbols that need to be inserted, deleted or substituted in order to transform the string in one language into the corresponding string in another language. In the simplest form of the algorithm, all operations have the same cost. We use 0 for the cost of mapping a character to itself, e.g. *a*:*a*, 1 to map it to a different character, e.g. *a*:*o*. Insertions and deletions of different characters cost 1. In more sensitive versions, base and diacritic may be distinguished. For example, the base of *ë* is *e*, and the diacritic is the diaeresis. Though it is not exactly clear what weight should be attributed to each of the components (Gooskens and Heeringa 2004), it is generally assumed that differences in the base will usually confuse the reader to a much greater extent than diacritical differences (Heeringa et al. 2013). If two characters have the same base but differ in diacritics, we assign them a substitution cost of 0.5.[7] In order to obtain distances which are based on linguistically motivated alignments, the algorithm is adapted so that in the alignment a letter representing a vowel (henceforth called a vowel letter) may only correspond to a vowel letter and a consonant letter only to a consonant letter.

We consider the normalized Levenshtein distance with regard to the assumption that a segmental difference in a word of two segments has a stronger impact on intelligibility than a segmental difference in a word of ten segments (Beijering et al. 2008). The normalized Levenshtein distance of BG–RU: *риба–рыба* (*riba–ryba*) *'fish'* is 1:4=0.25 or 25%. We calculate the average of the Levenshtein distance between stimuli of selected Slavic languages and their cognates in RU (Table 2). The assumption is: The larger the distance, the more difficult it is to comprehend the related language (Section 4.1).

Table 2. Normalized Levenshtein distances between Russian and other Slavic languages given as percentages

Slavic languages	RU
UK	23.87%
BE	28.92%
BG	25.61%
MK	28.92%
SR	34.26%

[7] Since we do not have any MK–RU cognates with the following alignment: *ѓ:г* and *ќ:к*, we weigh these pairs as 1 in our Levensthein distance matrix.

3.4. Word adaptation surprisal

In addition to the Levenshtein distance we use the information-theoretic concept of *surprisal*. The term *surprisal* was introduced by Tribus (1961), who used it to talk about the logarithm of the reciprocal of a probability (Hale 2016). Surprisal allows us to characterize the information value of an observed event and has been shown to correlate in many cases with various metrics of success, such as reading times in eye-tracking experiments (Boston et al. 2008, Smith and Levy 2013). Surprisal is defined as the code length of an optimal prefix-free code for a given probability distribution and thus has an inverse logarithmic relation to the probability values themselves (Shannon 1948). Surprisal values are given in bits and depend heavily on the used probability distribution.

We calculated the letter adaptation surprisal with the following formula (1). Letter adaptation surprisal values allow for quantifying the unexpectedness both of individual letter correspondences and of whole cognate pairs.

(1) $$surprisal(L1 = l1|L2 = l2) = -\log P(L1 = l1|L2 = l2)$$

L1 — native (decoder) language, l1 — letter of the native (decoder) language,
L2 — foreign (stimulus) language, l2 — letter of the foreign (stimulus) language

We can also compute the adaptation surprisal for string correspondences in our set. For example, the BG–RU cognate pair *глад–голод* (*glad–golod*) ‘*hunger*’ contains a string correspondence *ла*:*оло*. The adaptation surprisal of the string correspondence can be calculated by summing up the letter adaptation surprisal of the contained letters: 2.0506 surprisal for *ø*:*о*, 0.0 surprisal for *л*:*л* and 1.8210 surprisal for *а*:*о*.[8]

In this study we investigate the word adaptation surprisal. We compute full word adaptation surprisal by summing up the letter adaptation surprisals. For example, the BG–RU cognate pair *син–сын* (*sin–syn*) ‘*son*’ contains the correspondences *с*:*с* (0.0 surprisal), *и*:*ы* (0.6919 surprisal) and *н*:*н* (0.0 surprisal). Thus, the BG *син* (*sin*) ‘*son*’ has a word adaptation surprisal of 0.6919 bits for Russian readers. This gives a quantification of the (un)expectedness of the correct cognate in Russian *сын* (*syn*) ‘*son*’. As in the case with the Levenshtein distance, we also normalize the full word adaptation surprisal, e.g. 0.6919:3=0.2306 or 23.06% for BG–RU *син–сын* (*sin–syn*) ‘*son*’. We calculate the average value of the word adaptation surprisal between stimuli of selected Slavic languages and their cognates in Russian (Table 3): The higher the surprisal, the more difficult it is to comprehend the related language and the more time is needed to complete the translation task (Section 4.2 and 4.3).

[8] The letter adaptation surprisal values are calculated on the basis of 120 Bulgarian stimuli and their Russian cognates.

Table 3. Normalized word adaptation surprisal between Russian and other Slavic languages given as percentages

Slavic languages	RU
UK	34.79%
BE	48.05%
BG	50.30%
MK	73.01%
SR	79.55%

4. Results

4.1. Levenshtein distance and intelligibility score

To investigate the relationship between intelligibility and Levenshtein distance scores, the results of the orthographic intelligibility tests are correlated with the overall Levenshtein distances (Fig. 1).

There is a negative correlation of −0.89 ($p < 0.05$, $R^2 = 0.79$). In general, the orthographic intelligibility can be predicted well from the overall Levenshtein distances (the larger the distance, the more difficult it is to understand the related language). However, e.g. the BE–RU and MK–RU Levenshtein distances are equal (28.92%), but the intelligibility scores are different, e.g. BE–RU: 71.66% and MK–RU: 61.81%.

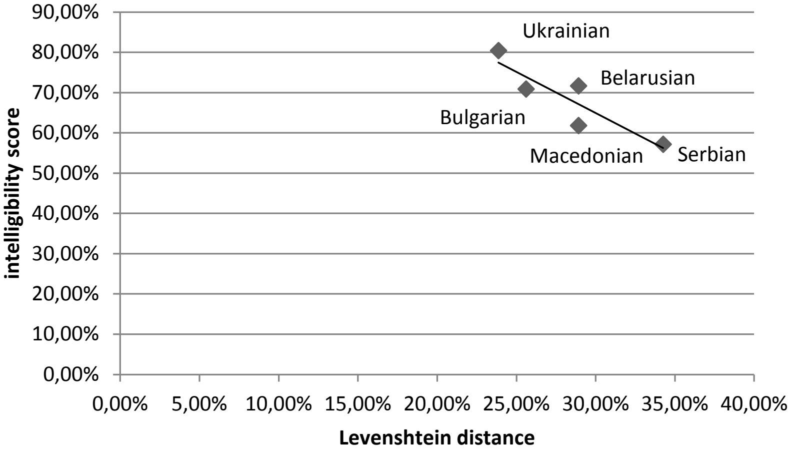


Fig. 1. The correlation between the mean Levenshtein distance and the average of the intelligibility score ($r = -0.89$)

4.2. Word adaptation surprisal and intelligibility score

The correlation between the mean word adaptation surprisal and the average of the intelligibility score (Fig. 2) shows that orthographic intelligibility can be predicted quite reliably ($r = -0.99$, $p < 0.001$, $R^2 = 0.98$) from the overall normalized word adaptation surprisal (the higher the value of surprisal the more difficult it is to comprehend the related language).

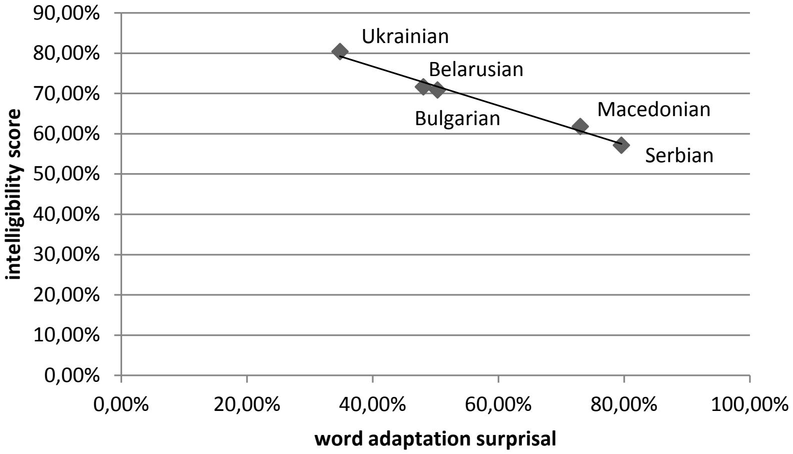


Fig. 2. The correlation between the mean word adaptation surprisal and the average of the intelligibility score ($r = -0.99$)

4.3. Word adaptation surprisal and total time

In addition to the mean intelligibility score we also correlate the normalized word adaptation surprisal with the total time spent per word incl. the initial time, typing time and final hesitation time (Fig. 3). The assumption was that people complete a translation task slower in the case of words with a higher surprisal value. In general, this assumption can be seen as confirmed ($r = 0.87$, $p = 0.05$, $R^2 = 0.76$). The scatterplot in Fig. 3 shows an interesting finding: Though the mean word adaptation surprisal between BE and RU (48.05%) is slightly lower than between BG and RU (50.30%) the total time between BE and RU (5.17 sec.) is slightly higher than between BG and RU (5 sec.). However, the mean value of intelligibility score between BE and RU (72.41%) is slightly higher than between BG and RU (70.88%) (Table 1). This means that participants were more successful in translating from BE into RU than from BG, but they spent slightly more time per word. The same situation concerns the MK–RU (5.72 sec.) and SR–RU (5.38 sec.) pairs. Though participants were more successful in translating from MK into RU than from SR, they spent more time per MK word.

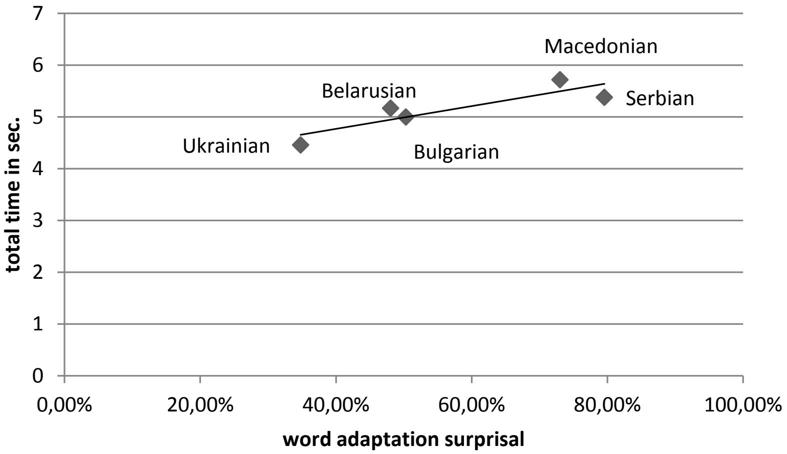


Fig. 3. The correlation between the mean word adaptation surprisal and the average of total reading and translation time (r = 0.87)

5. Conclusions and outlook

The aim of this article was to validate the normalized Levenshtein distance and the normalized word adaptation surprisal by investigating the degree to which these measurements of orthographic distances between Russian and five other Slavic languages, all using the Cyrillic alphabet, correlate with the mean intelligibility score and the total time obtained from Russian readers in web-based experiments. The results suggest that the word adaptation surprisal is a better predictor of orthographic intelligibility than the Levenshtein distance. We see this as a confirmation of the usefulness of the surprisal method. However, the difference is not significant between Levenshtein distance and word adaptation surprisal (r = −0.89 versus r = −0.99 respectively). The assumption that people complete a translation task slowly on words whose surprisal value is higher could be confirmed only partly. A possible explanation for this is that participants need more time for the cognitive effort required to process the information and to complete the task correctly.

The Levenshtein distance has often been used as a predictor of mutual intelligibility between related languages in spoken semi-communication. We decided to add the method employing the notion of surprisal in order to test its applicability in our scenario. Both methods have their advantages and disadvantages. The Levenshtein algorithm measures the distance between two cognates within a language pair: nonidentical correspondences contribute to the orthographic distance, identical ones do not. Nonidentical correspondences are regarded as different and cost 1 unit. The surprisal method measures the complexity of a mapping, more precisely, how predictable the correspondence is in a language pair. The surprisal values of correspondences are different. However, they depend on frequency and distribution of correspondences in the particular cognate set. Furthermore, surprisal can be asymmetrical: the surprisal values between language A and language B are not necessarily the same as between language B and language A. This indicates an advantage of the surprisal-based method compared to the Levenshtein distance, which in its basic form is completely symmetrical.

Focusing on orthographic intelligibility, orthographic correspondences themselves, as well as their frequency, their nature or their position can be expected to perform well as predictors of intelligibility (Stenger et al. forthcoming). In future research, using the refined Levenshtein distance and adaptation surprisal models, we will analyze mismatched orthographic correspondences more precisely in order to investigate what kind of correspondences either facilitate or hinder intercomprehension as well as to get qualitatively significant results.

The results of our study are relevant for the areas of written intelligibility as well as of spoken semi-communication. The way in which we tested intelligibility may be relevant for research in other experimental disciplines within the humanities such as psycholinguistics and education science. Furthermore, the adaptation surprisal method can be used to also measure phonetic/phonological as well as morphological intelligibility between related languages.

Acknowledgement: We wish to thank Andrea Fischer and Varvara Obolonchykova for their support in the calculation of the Levenshtein distances and word adaptation surprisal (based on Python 2.7.6).

References

1. *Beijering K., Gooskens C., Heeringa W.* (2008), Predicting intelligibility and perceived linguistic distance by means of the Levenshtein algorithm, Linguistics in the Netherlands, pp. 13–24.
2. *Bidwell C. E.* (1963), Slavic Historical Phonology in Tabular Form, The Hague: Mouton & Co.
3. *Boston M. F., Halle J., Kliegl R., Patil U., Vasishth S.* (2008), Parsing costs as predictors of reading difficulty: an evaluation using the Potsdam Sentence Corpus, Journal of Eye Movement Research 2 (1), pp. 1–12.
4. *Braunmüller K., Zeevaert L.* (2001), Semicommunication, receptive multilingualism and related phenomena. A bibliographical overview, [Semikommunikation, rezeptive Mehrsprachigkeit und verwandte Phänomene. Eine bibliographische Bestandaufnahme], Working papers in multilingualism [Arbeiten zur Mehrsprachigekeit], Series B, No. 19, University Hamburg [Universität Hamburg].
5. *Carlton T. R.* (1991), Introduction to the Phonological History of the Slavic Languages, Slavica Publishers, Inc., Columbus, Ohio.
6. *Doyé P.* (2005), Intercomprehension. Guide for the development of language education policies in Europe: from linguistic diversity to plurilingual education. Reference study, Strasbourg, DG IV, Council of Europe.
7. *Fischer A., Jágrová K., Stenger I., Avgustinova T., Klakow D., Marti R.* (2015). An Orthography Transformation Experiment with Czech-Polish and Bulgarian-Russian Parallel Word Sets, in Sharp B., Lubaszewski W., Delmonte R. (eds.), Natural Language Processing and Cognitive Science 2015 Proceedings, Libreria Editrice Cafoscarina, Venezia, pp. 115–126.

8. *Golubović J., Gooskens, C.* (2015), Mutual intelligibility between West and South Slavic languages, Russ Linguist 39, Springer, pp. 351–373.
9. *Gooskens C., Heeringa W.* (2004), Perceptive evaluation of Levenshtein dialect distance measurements using Norwegian dialect data, Language Variation and Change, 16, Cambridge University Press, pp. 189–207.
10. *Gooskens C.* (2006), Linguistic and extra-linguistic predictors of Inter-Scandinavian intelligibility, in van de Weijer J. and Los B. (eds.), Linguistics in the Netherlands, 23, John Benjamins, Amsterdam, pp. 101–113.
11. *Gooskens C.* (2007), The contribution of linguistic factors to the intelligibility of closely related languages, Journal of Multilingual and Multicultural Development 28(6), pp. 445–467.
12. *Gooskens C.* (2013), Experimental methods for measuring intelligibility of closely related language varieties, in Bayley R., Cameron R., Lucas C. (eds.), Handbook of Sociolinguistics, Oxford University Press, Oxford, pp. 195–213.
13. *Halle J.* (2016), Information-theoretical Complexity Metrics, Language and Linguistics Compass 10/9, pp. 397–412.
14. *Haugen E.* (1966), Semicommunication: The language gap in Scandinavia, Sociological Inquiry 36, pp. 280–297.
15. *Heeringa W., Golubovic J., Gooskens C., Schüppert A., Swarte F., Voigt S.* (2013), Lexical and orthographic distances between Germanic, Romance and Slavic languages and their relationship to geographic distance, in Gooskens C. and van Bezoijen R. (eds.), Phonetics in Europe: Perception and Production, Peter Lang, Frankfurt a.M., pp. 99–137.
16. *Ivanova V. F.* (1991), Modern Russian orthography [Sovremennaja russkaja orfografija], Vysšaja škola, Moskva.
17. *Kessler B.* (1995), Computational dialectology in Irish Gaelic, Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, Dublin: EASCL, pp. 60–67.
18. *Kučera K.* (2009), The Orthographic Principles in the Slavic Languages: Phonetic/Phonological, in Kempgen S., Kosta P., Berger T., Gutschmidt K. (eds.), The Slavic Languages. An International Handbook of their Structure, their History and their Investigation, Vol. 1. Walter de Gruyter, Berlin & New York, pp. 70–76.
19. *Kürschner S., van Bezooijen R., Gooskens C.* (2008), Linguistic determinants of the intelligibility of Swedish words among Danes, International Journal of Humanities and Arts Computing 2(1/2), pp. 83–100.
20. *Levenshtein V. I.* (1965), Binary codes capable of correcting deletions, insertions, and reversals [Dvoičnye kody s ispravleniem udalenij, vstavok i zamen simvolov], Doklady of the Soviet Academy [Doklady Akademii Nauk SSSR], 1965, Vol. 163, No. 4, pp. 845–848.
21. *Marti R.* (2014), Historical Graphemics of the Slavic Languages: the Glagolitic and Cyrillic Writing Systems [Historische Graphematik des Slavischen: Glagolitische und kyrillische Schrift], in Kempgen S., Kosta P., Berger T., Gutschmidt K. (eds.), The Slavic Languages. An International Handbook of their Structure, their History and their Investigation, Vol. 2. Walter de Gruyter, Berlin & New York, pp. 1497–1514.

22. *Mel’ničuk O.S.* (1966), Introduction to comparatively-historical studies of Slavic languages [Vstup do porivnjal’no-istoryčnoho vyvčennja slov‘'jans’kich mov], Naukova dumka, Kiev.
23. *Möller R., Zeevaert L.* (2015), Investigating word recognition in intercomprehension: Methods and findings, Linguistics 2015 53(2), De Gruyter Mouton, Berlin, Munich & Boston, pp. 313–352.
24. *Musatov V. N.* (2012), Russian language. Phonetics, Phonology, Orphoepy, Graphics, Orthography [Russkij jazyk. Fonetika, Fonologija, Orfoėpija, Grafika, Orfografija], Izdatel’stvo ‘Flinta’, Moskva.
25. *Sgall P.* (2006), Towards a Theory of Phonemic Orthography, in Sgall P. (ed.), Language in its multifarious aspects, Karolinum Press Charles University, pp. 430–452.
26. *Shannon C. E.* (1948), A mathematical theory of communication, Bell System Technical Journal 27 (379–423), pp. 623–656.
27. *Smith N. J., Levy R.* (2013), The effect of word predictability on reading time is logarithmic, Cognition 128(3), pp. 302–319.
28. *Stenger I.* (2016), How Reading Intercomprehension Works among Slavic Languages with Cyrillic Script, in Köllner M., Ziai R. (eds.), Proceedings of the ESSLLI 2016, pp. 30–42, available at: http://esslli2016.unibz.it/wp-content/uploads/2016/09/esslli-stus-2016-proceedings.pdf
29. *Stenger I., Jágrová K., Fischer A., Avgustinova T.* (Forthcoming), “Reading Polish with Czech Eyes” or “How Russian Can a Bulgarian Text Be?”: Orthographic Differences as an Experimental Variable in Slavic Intercomprehension, in Kosta P., Radeva-Bork T. (eds.), (preliminary title) Current developments in Slavic Linguistics. Twenty years after, Peter Lang.
30. *Tribus M.* (1961), Thermostatics and thermodynamics, D. van Nostrand Company.
31. *Vanhove J.* (2015), The Early Learning of Interlingual Correspondences Rules in Receptive Multilingualism. International Journal of Bilingualism, available at: http://homeweb.unifr.ch/VanhoveJ/Pub/papers/Vanhove_Correspondence-Rules.pdf. Vanhove J., Berthele R. (2015), The lifespan development of cognate guessing skills in an unknown related language, International Review of Applied Linguistics in Language Teaching 53(1), pp. 1–38.
32. *Vasmer M.* (1973), Etymological dictionary of the Russian language [Ėtimologičeskij slovar’ russkogo jazyka], Progress, Moskva.
33. *Žuravlev A. F.* (ed.) (1974–2012), Etymological dictionary of the Slavic inherited lexikon. Proto-Slavic lexical stock [Ėtimologičeskij slovar’ slavjanskich jazykov. Praslavjanskij leksičeskij fond], Vol. 1–37. Nauka, Moskva.

COREFERENCE RESOLUTION IN RUSSIAN: STATE-OF-THE-ART APPROACHES APPLICATION AND EVOLVEMENT

Sysoev A. A. (sysoev@ispras.ru),
Andrianov I. A. (ivan.andrianov@ispras.ru),
Khadzhiiskaia A. Y. (sanya@ispras.ru)

Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia

Coreference resolution aims at grouping textual references denoting same real world entities into clusters. Many state-of-the-art results have already been received for coreference resolution in European languages, but for Russian this area is still quite novel and underexplored. With this paper we try to fill this gap. Our article reviews existing approaches and presents their adaptation for Russian language. We carry out sufficient number of experiments to estimate efficiency of various machine learning methods and features, utilized under the hood of the algorithms. Additionally we propose a novel feature to be used for head detection subtask, which is based on word embeddings clustering. As a result, we managed to establish baseline implementation for Russian language coreference resolution problem. The key features of the developed approach are simplicity and extensibility. Presence of such a baseline opens many research directions for improving quality of the algorithms; some potential improvements are already pointed out in this paper. We expect further works in this area to significantly increase current level of state-of-the-art results for Russian coreference resolution, making it practically applicable in the near future.

Keywords: coreference resolution, anaphora resolution, mention detection, natural language processing, machine learning, word embeddings

РАЗРЕШЕНИЕ КОРЕФЕРЕНТНОСТИ: ПРИМЕНЕНИЕ И РАЗВИТИЕ СОВРЕМЕННЫХ ПОДХОДОВ

Сысоев А. А. (sysoev@ispras.ru),
Андрианов И. А. (ivan.andrianov@ispras.ru),
Хаджийская А. Ю. (sanya@ispras.ru)

Институт системного программирования
Российской академии наук, Москва, Россия

1. Introduction

Coreference resolution aims at grouping natural language expressions into clusters, according to entities of the real world, they denote. As text is naturally linear, such clusters are usually represented as chains or trees. The mention, which already has some meaning, is called antecedent; while the mention, which borrows its meaning from antecedent, is called anaphor.

There are many research projects, targeting coreference resolution problem for European languages; many state-of-the-art results have already been received. However, for Russian language this area is quite novel and the modern state can hardly be clearly defined. In this article we try to fill this gap. We review approaches commonly used to solve coreference resolution problem for foreign languages and adapt these approaches for Russian.

Thus the main contribution of the paper is threefold. First of all we apply state-of-the-art methods for coreference resolution in European languages (especially English) to Russian. We also provide detailed analysis of different algorithms and features usefulness. Additionally we present a novel feature based on word embeddings clustering for one of the building blocks—head detection algorithm.

2. Related work

Coreference resolution started its long history with initial attempts to resolve pronoun references [6]. It experienced another development in the mid 1990s, when several specific coreference resolution tasks were issued during Message Understanding Conferences. In 2001 there appeared a fundamental work [15], presenting a machine learning approach to building coreference resolution algorithms. The proposed method started from generating a number of entity mentions. Then each potential antecedent-anaphor pair was classified by pre-trained decision tree. Finally, mention pairs, classified as coreferent, were aggressively merged into clusters, each representing a separate entity.

Another epoch in coreference resolution started with ConLL-2011/2012 shared tasks for modeling unrestricted coreference, which provoked a number of novel approaches. [9] presented a multi-filter method, where each level established or forbade links between pairs of mentions. Idea presented in [15] with mention pairs classification further evolved in [4] with more features and machine learning algorithms being used. [5,11] utilized more sophisticated approaches to coreference resolution, regarding the task as global optimization for all document mentions, in contrast to local optimization for selected mention pairs. They managed to gain state-of-the-art results on ConLL datasets.

However, [25] claimed that even simple mention-pair classification algorithms can achieve top-level results. It proposed two main improvements: easy-first mention-pair clustering algorithms, utilizing not only positive classification predictions, but also negative ones as non-grouping constraints; additionally it exploited Jaccard Item Set mining [14] feature selection to inject non-linear features into linear predictor.

Recent experiments with coreference resolution in Russian were conducted as a part of RU-EVAL 2014 evaluation campaign [17]. For its purposes the first Russian coreference corpus was compiled and manually annotated [18].

The evaluation track consisted of two tasks: coreference chains identification and anaphora resolution. Organizers reported three participants in coreference resolution task; however, they did not provide any results of the evaluation and the papers published on the matter covered mostly anaphora resolution, see [2, 7, 8, 12]. Only the first article suggested a number of rule-based techniques to resolve mentions coreferring with named entities but unfortunately they did not evaluate proposed methods.

Another research on Russian coreference was presented in [19]. This article described a machine learning based system. However, authors focused on the description of two experimental modules for sieving singleton and anaphoric mentions and did not provide detailed information about their resolver. The overall quality of their algorithms showed F1-measure of 48.04% on MUC metric [26] and 32.51% on B3 metric [1].

3. Method description

In this section we describe our method for building clusters of coreferent mentions for input text.

Our method consists of two main steps: mention detection and coreference resolution. Mention detection algorithm extracts word expressions that are possible elements of coreference chains. Extracted mentions are further grouped into coreferent clusters.

The first step of mention detection is further divided into two parts: collecting all mention heads in the given document and expanding them to full mentions. Such two-stage algorithm is inspired by [11], though the very definition of mention head in our implementation differs from the one provided in the paper.

Our algorithm requires text documents to be preprocessed, which includes morphological analysis (part of speech tags, grammemes, lemmas), syntactic analysis (dependency trees) and named entity recognition. All these steps are performed by Texterra system [20].

Additionally we train lemma-based word embeddings (skip-gram word2vec vectors with 50 dimensions) on RuEval-2014 corpus [17], FactRuEval-2016 test corpus [16], Russian section of Wikinews[1] and internal newswire corpora.

3.1. Head detection

Our system is designed to resolve entity coreference by establishing links between noun phrases and quantified phrases. Thus we consider a mention head to be a single token tagged as either noun, or numeral, or (as an exception) adjective pronoun (possessive, relative or demonstrative). We view the task of heads identification as a binary classification problem aiming to distinguish candidates as true/false mention heads. We employ a number of heuristics to obtain sure heads, which do not require further classification. Pronouns are anaphoric by definition and therefore are always incorporated in some coreferent chain. We also interpret named entities annotated by pre-processing tool as sure mentions, thus their heads are added heuristically to the resulting

[1] https://ru.wikinews.org/

set. We follow named entities restriction on overlapping. All non-head nouns and numerals nested in named entities are skipped during candidate head generation step.

Our feature set for classified candidate tokens can be divided into several groups each capturing linguistic insights on various levels of language organization. Linguistic factors behind our feature set are backed up by a number of papers focused on categorizing discourse entities in light of their role in the coreferent text structure: anaphoric expressions, singletons, antecedents, etc [3, 11, 13, 23].

Internal morphological features include basic information about token such as POS-tag and corresponding grammemes: number, gender, animacy, etc. Syntax group encodes position and relations of a token within a sentence, representing candidate local salience. Syntactic context features contain morphological features for syntactic parent deduced from a dependency tree. Context group includes the same basic morphological features for two left and right token neighbours. Frequency feature set consists of tf weighting for both word form and lemma.

Semantic features utilize pre-trained word embeddings. Lemma vectors for mention heads in the training corpora are clustered by KMeans++, then cosine similarity and Euclidean distance between candidate token lemma and each of given cluster centers are exposed as features. The intuition is that each cluster represents a “sense” and high similarity (low distance) to any “sense” is highly correlated to being head. Described technique allows to attend to data sparsity problem present in most NLP tasks by recognizing heads that are not present in the training corpora but are similar to ones that are.

3.2. Head expansion

Our head expansion method is partially similar to [11]. To determine left (right) mention boundary we iterate through tokens to the left (right) starting from the nearest neighbour of mention head and apply a binary classifier until it predicts *false* label. The final token on which classifier predicts *true* label is a mention boundary. There is a simple exception for the given method: pronoun heads are treated as full mentions without any classification.

Classifier features can be categorized as following: token-based, position-based, context-based. Token-based features include word form, lemma and part-of-speech information about head/candidate token and its nearest neighbours. Position-based features include direction from head to candidate, distance between head and candidate and whether head/candidate is the first/last token of the sentence. Context-based features reflect whether head and candidate are parts of the same named entity, whether head/candidate is a syntactic ancestor (in terms of a dependency tree) of the other one and part-of-speech pattern for words between head and candidate.

3.3. Coreference resolution algorithm

Given a collection of mentions as input coreference resolution algorithm clusters them into groups, each denoting a single entity. Our approach is highly inspired by [25]: we start from generating a number of mention pairs, which are then classified as belonging to one coreference cluster or not. Additionally, the classifier provides some confidence estimation of its decision. Finally, analyzed mention pairs are merged into clusters according to Easy-First Mention-Pair approach presented in [25].

3.3.1. Generation of Candidate Mention Pairs

Generation of candidate mention pairs occurs during two phases: training and testing.

In training the standard method [15, 25] is to construct positive examples from nearest valid coreferent pairs; each mention between members of correct pair coupled with pair's anaphor delivers negative examples.

In testing phase a window specifying a number of neighbor mentions to be taken from the left side of each text mention is usually applied. In our approach we choose the window size covering distances between mentions of 98% valid coreferent pairs within training corpus.

3.3.2. Mention pair classification

All mention pairs are converted into feature vectors, which are then classified. Populated feature vectors consist of several groups:

- **Basic linguistic features** include word forms [5, 25], lemmas, part-of-speech tags [5] and grammemes (gender, number, animacy) [4] for mention head and context words. Context is composed from up to two same sentence tokens to the left and to the right of the considered mention [5, 25].
- **Grammemes agreement features** are indicators of mention heads sharing the same key grammemes (number [15, 24, 25], gender [4], animacy [24, 25], pronominality [5]).
- **Positional features** provide information about mentions arrangement in text: distance [5,15] and place within sentence boundaries [24,25].
- **Named entity based features** provide information about mention types and their agreement [5, 24, 25].
- **Structural features** encode information about mention size [5, 25] and interrelation with other mentions of the text (intersecting [22]; containing other mentions or being contained in other mentions [5, 24, 25]).
- **Surface form matching features** include lexicographic similarity [21] and textual representation equality indicators [5, 15, 24, 25]. In our approach features of this group are based on lemmas, constituting mentions, rather than on their word forms.
- **Syntactic features** incorporate information, that can be extracted from dependency trees, such as grammar role, sharing same parent node or clause and so on [22].

Additionally, conforming to [25], in conjunction with linear classifier we use Jaccard Item Set mining algorithm [14] to gain sets of features, frequently appearing together. Joining these features into a single composite provides the ability to utilize even primitive features, targeting only one mention of the classified pair.

3.3.3. Easy-First Mention-Pair algorithm

Easy-First Mention-Pair algorithm [25] receives a number of candidate mention pairs together with their classification results—whether the pair is valid or not and confidence of this prediction. Provided pairs are sorted by confidence, so that more precisely classified pairs come first. Initially, each mention is assigned to its own cluster. Confidence-ordered list of mention pairs is walked down sequentially: pairs, classified as valid, are merged into a single cluster; pairs, classified as invalid, are memorized as unlinking constraint to prevent merging further pairs with lower confidence. If merging two clusters results in one containing a previously unlinked pair the analyzed pair is just ignored. Clusters present after full list traversal represent target groups of coreferent mentions.

4. Evaluation

In this section we describe corpus used for testing and present detailed evaluation results for each of the algorithm steps.

4.1. RuEval-2014 corpus

We employ the RuCor coreference corpus [17] as a training and evaluation set for all subtasks described above. It contains 181 texts (about 200,000 tokens) representing five written genres: news, essays, fiction, scientific articles and blog posts.

During our experiments we encountered two major problems in the original dataset that required manual fixes: duplicated mentions and cyclic chains. The first problem was caused by erroneous merging of markup variants from different annotators. In order to avoid merging discrepancies we took into consideration only mentions with the most popular variant label “1”. Two documents from the original dataset contained no variants with this label and had to be dismissed. We also detected and manually straightened 6 chains containing cycles.

Another problem with the dataset arose from different definitions of mention head. Some tokens that could serve as a head for potential mention were not annotated. Sometimes for the purposes of evaluation campaign annotators marked several heads in dubious cases, such as appositional proper names and coordinate noun phrases, 1,696 multi-token heads in total. Following the constraints of dependency parsing our system expects head to be a single token. In order to process these cases we applied simple heuristic taking the first noun as a correct head.

These inconsistencies make it difficult to evaluate absolute quality of the algorithms with RuCor dataset, but comparative analysis appears reasonable.

4.2. Head detection

We employ groundtruth heads retrieved from corpus to test the quality of our head detection algorithm in 10-fold cross-validation applying standard metrics such as precision, recall and F1-measure. We invoke our detector with and without semantic features (based on word embeddings) to evaluate impact of this feature group on the overall quality and make several runs to find the optimal number of clusters (Table 1).

Table 1. Head detection evaluation

Setting: logistic regression with L2 regularization	Precision	Recall	F1-measure
without semantic features	**0.7326**	0.6628	0.6952
with 105 clusters	0.7289	0.6839	0.7050
with 110 clusters	0.7288	0.6841	0.7051
with 115 clusters	0.7289	**0.6842**	**0.7052**
with 120 clusters	0.7280	0.6839	0.7046
with 125 clusters	0.7288	0.6838	0.7049

Table 1 shows that semantic features are highly influential and can dramatically increase recall and F1-measure. The best general quality is reached with 115 clusters therefore we use this number in the full pipeline as an empirically preset variable.

4.3. Head expansion

Given gold mention heads we evaluate our head expansion algorithm in 10-fold cross-validation with three information retrieval metrics: precision, recall, F1-measure. We additionally provide ablation analysis results for context-based features, as their usefulness is most controversial (Table 2).

Table 2. Head expansion evaluation

Setting: logistic regression with L2 regularization	Precision	Recall	F1-measure
all	0.8682	0.8654	0.8668
all—same NE	0.8652	0.8627	0.8640
all—syntactic ancestor	0.8483	0.8440	0.8461
all—POS pattern	**0.8687**	**0.8660**	**0.8673**
all—context features	0.8441	0.8401	0.8421

As we can see the most influential context-based feature is syntactic ancestor. This looks reasonable as mentions are generally expected to be subtrees of the sentence syntactic tree. The worst context-based feature is POS pattern, it even introduces minor loss in all metrics, which requires additional analysis.

4.4. Coreference resolution

MUC [26], B3 [1], $\text{CEAF}_{\text{entity}}$ and $\text{CEAF}_{\text{mention}}$ [10] versions of precision, recall and F1-measure are used to evaluate performance of coreference resolution approach in 10-fold cross-validation. Experiments within this section are carried out with ground-truth mentions, thus $\text{CEAF}_{\text{mention}}$ metrics are all the same.

First of all, we examine impact of different classification algorithms utilized under the hood of coreference resolution (Table 3).

Table 3. Evaluation of machine learning algorithms in coreference resolution

Metric \ Machine learning algorithm		**logistic regression**	**logistic regression + Jaccard Item Set mining**	**random forest**
MUC	Precision	0.7246	0.7333	**0.7395**
	Recall	0.6969	**0.7027**	0.6520
	F1	0.7104	**0.7175**	0.6928
B3	Precision	0.5852	0.6014	**0.7389**
	Recall	**0.6104**	0.6103	0.5516
	F1	0.5973	0.6055	**0.6312**
$\text{CEAF}_{\text{mention}}$	Precision/Recall/F1	0.5284	0.5375	**0.5894**
$\text{CEAF}_{\text{entity}}$	Precision	0.4765	**0.4825**	0.4780
	Recall	0.5396	0.5514	**0.6583**
	F1	0.5057	0.5140	**0.5533**

As it can easily be seen, Jaccard Item Set mining algorithm, introducing non-linear features into logistic regression, slightly improves its quality. However, random forest manages to show even better progress. This result looks reasonable, as decision trees within random forest algorithm are naturally designed to induce knowledge from combinations of even most trivial features. Jaccard Item Set mining should have performed similarly, however it is probable that we did not succeed to choose optimal parameters for it.

To determine most significant features for coreference resolution task ablation analysis experiments (with random forest classifier) are carried out (Fig. 1, 2).

Basic linguistic and syntactic features seem useless for coreference resolution tasks—removing them might even slightly improve results, while surface form features are the most valuable ones.

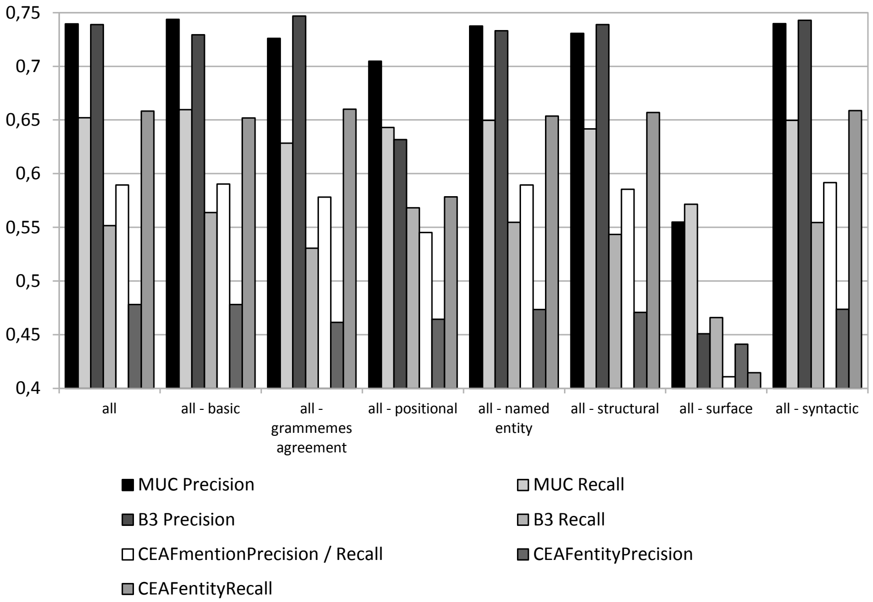


Fig. 1. Ablation analysis for coreference resolution. Precision and recall

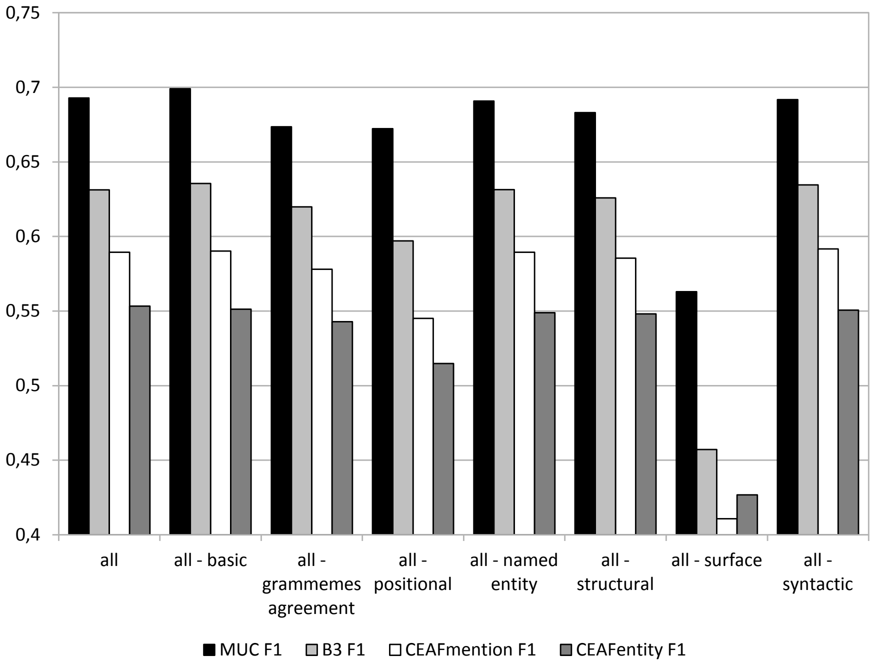


Fig. 2. Ablation analysis for coreference resolution. F1

4.5. Full coreference resolution pipeline

This section presents the results of evaluating coreference pipeline as a whole—starting from head detection and expansion. During each iteration of 10-fold cross-validation parts of the algorithm were sequentially trained and then applied to testing documents. Final results are presented in Table 4.

Table 4. Evaluation results of full coreference resolution pipeline

Precision	Recall	F1	Precision	Recall	F1
MUC			B3		
0.4768	0.3741	0.4189	0.4104	0.2957	0.3431
$\mathrm{CEAF}_{\mathrm{mention}}$			$\mathrm{CEAF}_{\mathrm{entity}}$		
0.4024	0.3702	0.3854	0.2525	0.3433	0.2906

These results significantly differ from the scenario with ground-truth mentions which is explained with accumulation of errors of all intermediate steps.

5. Conclusion

In this paper we aimed to evaluate the usefulness of coreference resolution approaches, developed for European languages, when applied to Russian. Presumably, we accomplished some baseline implementation for this problem. The key features of the developed approach are simplicity and extensibility, which opens many research lines in this area. We consider the following directions to be beneficial in the near future:

- carrying out experiments with more machine learning algorithms and approaches;
- using various clustering algorithms for word embeddings;
- detailed analysis of features, assumed useless in ablation experiments;
- tuning coreference resolution algorithm for different mention types.

References

1. *Bagga A., Baldwin B.* (1998), Algorithms for Scoring Coreference Chains, The first international conference on language resources and evaluation workshop on linguistics coreference, pp. 563–566.
2. *Bogdanov, A. V., et al.* (2014), Anaphora analysis based on ABBYY Compreno linguistic technologies, Computational Linguistics and Intellectual Technologies, issue 13, pp. 89–102.
3. *De Marneffe, M.-C., Recasens M., Potts C.* (2015), Modeling the lifespan of discourse entities with application to coreference resolution, Journal of Artificial Intelligence Research, iss. 52, pp. 445–475.
4. *Dos Santos C. N., Carvalho D. L.* (2011), Rule and Tree Ensembles for Unrestricted Coreference Resolution, Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 51–55.

5. *Fernandes E. R., Dos Santos C. N., Milidiú R. L.* (2012), Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution, Joint Conference on EMNLP and CoNLL-Shared Task, pp. 41–48.
6. *Hobbs J.* (1978), Resolving pronoun references, Lingua, vol. 44, issue 4, pp. 311–338.
7. *Ionov M., Kutuzov, A.* (2014), The impact of morphology processing quality on automated anaphora resolution for Russian, Computational Linguistics and Intellectual Technologies, issue 13, pp. 232–240.
8. *Kamenskaya M. A., Khramoin I. V., Smirnov I. V.* (2014), Data-driven methods for anaphora resolution of Russian texts, Computational Linguistics and Intellectual Technologies, issue 13, pp. 241–250.
9. *Lee H., Peirsman Y., Chang A., Chambers N., Surdeanu M., Jurafsky D.* (2011), Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task, Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pp. 28–34.
10. *Luo, X.* (2005), On Coreference Resolution Performance Metrics, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 25–32.
11. *Peng H., Chang K.-W., Roth D.* (2015), A Joint Framework for Coreference Resolution and Mention Head Detection, CoNLL, vol. 51, pp. 12–22.
12. *Protopopova E. V., Bodrova A. A., Volskaya S. A., Krylova I. V., Chuchunkov A. S., Alexeeva S. V., Bocharov V. V., Granovsky D. V.* (2014), Anaphoric Annotation and Corpus-based Anaphora Resolution: an experiment, Computational Linguistics and Intellectual Technologies, issue 13, pp. 562–571.
13. *Recasens M., De Marneffe M.-C., Potts C.* (2013), The Life and Death of Discourse Entities: Identifying Singleton Mentions, HLT-NAACL, pp. 627–633.
14. *Segond M., Borgelt C.* (2011), Item Set Mining Based on Cover Similarity, Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 493–505.
15. *Soon W. M., Ng H. T., Lim D. C. Y.* (2001), A Machine Learning Approach to Coreference Resolution of Noun Phrases, Computational linguistics, vol. 27, issue 4, pp. 521–544.
16. *Starostin A. S. et al.* (2016), FactRuEval 2016: Evaluation of Named Entity Recognition and Fact Extraction Systems for Russian, Computational Linguistics and Intellectual Technologies, issue 15, pp. 702–720.
17. *Toldova S. et al.* (2014), RU-EVAL-2014: Evaluating anaphora and coreference resolution for Russian, Computational Linguistics and Intellectual Technologies, issue 13, pp. 681–694.
18. *Toldova S., Ghrishina Y., Ladygina A., Vasilyeva M., Sim G., Azerkovich I.* (2016), Russian Coreference Corpus, Input a Word, Analyze the World: Selected Approaches to Corpus Linguistics, Cambridge Scholars Publishing, Newcastle Upon Tyne, pp. 107–124.
19. *Toldova S., Ionov M.* (2016), Mention Detection for Improving Coreference Resolution in Russian Texts: A Machine Learning Approach, Computación y Sistemas, vol. 20, issue 4, pp. 681–696.
20. *Turdakov D. Y. et al.* (2014), Texterra: A framework for text analysis, Programming and Computer Software, vol. 40, issue 5, pp. 288–295.

21. *Uryupina O.* (2004), Evaluating Name-Matching for Coreference Resolution, Proceedings of LREC, pp. 1339–1342.
22. *Uryupina O.* (2006), Coreference Resolution with and without Linguistic Knowledge, Proceedings of LREC, pp. 893–898.
23. *Uryupina O.* (2009), Detecting anaphoricity and antecedenthood for coreference resolution, Procesamiento del lenguaje natural, issue 42, pp. 113–120.
24. *Uryupina O., Moschitti A., Poesio M.* (2012) BART goes multilingual: The UniTN / Essex submission to the CoNLL-2012 Shared Task, Joint Conference on EMNLP and CoNLL-Shared Task, pp. 122–128.
25. *Uryupina O., Moschitti A.* (2015), A State-of-the-Art Mention-Pair Model for Coreference Resolution, Lexical and Computational Semantics (SEM 2015), pp. 289–298.
26. *Vilain M., Burger J., Aberdeen J., Connolly D., Hirschman L.* (1995), A model-theoretic coreference scoring scheme, Proceedings of the 6th conference on Message understanding, pp. 45–52.

COREFERENCE RESOLUTION FOR RUSSIAN: THE IMPACT OF SEMANTIC FEATURES[1]

Toldova S. (toldova@yandex.ru)
National Research University Higher School of Economics, Moscow, Russia

Ionov M. (max.ionov@gmail.com)
Lomonosov Moscow State University, Moscow, Russia

This paper presents the results of our experiments on building a general coreference resolution system for Russian. The main aim of those experiments was to set a baseline for this task for Russian using the standard set of features developed and tested for coreference resolution systems created for other languages. We propose several baseline systems, both rule-based and ML-based. We show that adding some semantic information is crucial for the task and even the small amount of data can improve the overall result. We show that different types of semantic resources affect the performance differently and sometimes more does not imply better.

Key words: coreference resolution, semantic features, ontologies, mention-pair coreference resolution, Russian NLP

РАЗРЕШЕНИЕ КОРЕФЕРЕНТНЫХ СВЯЗЕЙ ДЛЯ РУССКОГО ЯЗЫКА: ВКЛАД СЕМАНТИЧЕСКИХ ПРИЗНАКОВ

Толдова С. (toldova@yandex.ru)
НИУ Высшая Школа Экономики, Москва, Россия

Ионов М. (max.ionov@gmail.com)
МГУ им. Ломоносова, Москва, Россия

1. Introduction

An important task for a number of high-level NLP applications, such as machine translation, summarization and storyline detection is the task of coreference resolution grouping the noun phrases that are the mentions of the same referent into one cluster.

All the noun phrases in one cluster form a coreference chain. A noun phrase that is a part of a coreference chain is called mention.

[1] This research was supported by a grant from Russian Foundation for Basic Research Fund (15-07-09306).

Even though there was a lot of research connected to the task of coreference resolution in the last 3 decades, there is still much work to do. One of important research directions in this field in the last decade is applying this task to less-resourced languages (e.g. Polish ([14]), Basque ([17]), and Czech ([11])). After RuCor, the first open Russian corpus with coreference annotation, was made available to the public ([18]), it became possible to create a coreference resolution system for Russian.

2. Background

Although the research on coreference resolution started more than 40 years ago (some of the classical papers include [5]–[2]), the machine learning approach to this task is relatively recent. One of the first papers on applying the ML approach to the task of coreference resolution was the seminal paper of Soon et al. ([16]). It introduced the mention-pair model of coreference resolution which was widely used since then.

This model works as follows: for every noun phrase that could be a mention generates a number of candidate antecedents from the preceding noun phrases. For each pair the classifier is invoked. The first (or the best, depending on the algorithm) positive pair is chosen. A set of pairs for training the classifier is created in a similar way.

Even though this model has flaws, e.g. its locality (it allows incompatible pairs in the chain) it is still widely used, especially as a baseline model.

3. Experiments

3.1. Data

Our experiments were conducted on RuCor, a Russian coreference corpus initially created as a dataset for the RU-EVAL campaign[2]. The collection contains 180 texts or text fragments (3,638 coreferential chains with 16,557 noun phrases in total) taken from different genres, such as news, scientific articles, blog posts and fiction. The corpus is already preprocessed: each text is tokenized, split into sentences, morphologically tagged and syntactically parsed using the tools developed by Serge Sharoff ([15]). The morphological tags were checked and fixed manually, since it was previously shown that errors on this level affects significantly the quality of a related task ([6]).

The annotation scheme is based on MUC-6 scheme. It includes only the annotation of the expressions referring to the real-world entities (e.g. there are no coreference links for abstract notions or generic expressions). The other constraint is that only identity relation is annotated, bridging or near-identity relations were not taken into consideration. The Gold Standard corpus contains annotations only for those NPs that are mentions (i.e. parts of a coreference chains), so singletons are not annotated.

[2] The corpus may be downloaded from http://rucoref.maimbava.net.

For our experiments[3] the corpus was randomly split into a training and a test set (70% and 30% respectively). We performed experiments on both gold mentions NPs that are annotated in the corpus as parts of some coreference chain and predicted mentions all NPs extracted from the corpus automatically based on the dependency parses.

For testing we used CoNLL reference coreference scorers ([13]) a set of tools used for scoring in the CoNLL evaluation campaign as a reliable implementation of scoring algorithms that can produce comparable results. Particularly, we used two metrics to evaluate our experiments: the MUC score ([19]) and the B3 score ([1]). The former is a baseline score used in nearly every paper on coreference resolution. Even though it has some flaws (e.g. a baseline system that treats all mentions as one coreference chain achieves around 80% precision and 100% recall on a MUC-5 corpus), it is crucial to provide this score when establishing a baseline. The latter provides the quality of constructing coreference chains on average hence gives a good approximation of how well the method works in general.

The scores are calculated based on the full noun phrases, so the error in the NP extraction leads to decreasing the coreference score. In order to evaluate the coreference resolution step itself, without penalties for the incorrect NP extraction, we used the so-called *Gold boundaries* evaluation strategy: the edges of the noun phrases from the coreferent chains were corrected using the GS data.

3.2. Features for the baseline models

Initially we created a set of mention-pair classifiers using simple shallow features proposed in a seminal paper by Soon et al. ([16]), a system that is often used as a baseline for machine learning models for coreference resolution.

Some features used in the paper are inapplicable to Russian in a straightforward way, for example, the feature *Definite Noun Phrase*, which should be 1 if the noun phrase starts with a definite article. Given that Russian is an article-less language, detecting definite NPs is a separate, complicated task.

Some other features are hard to implement, for example *Semantic class agreement*, which should be set to 1 if the two candidate NPs has the same semantic class. Another feature like this is *Alias*, which should be set to 1 if one NP is an alias of another. Due to the small amount of available NLP tools and resources that work with Russian, there is no straightforward way to obtain values for those features. Available tools and possible ways to extract this knowledge are discussed in section 3.5.

To compensate this, in the baseline system we replaced those features with the heuristics which use other shallow features that should correlate with original missing ones:

1. *Animacy agreement*: True if both NP are animate or both NP are inanimate. This feature is used as a poor-man replacement for a *Semantic Class agreement* feature. The class hierarchy in this case consists of two classes on one level: *object / living thing*.

[3] The Jupyter notebooks which reproduce the experiments may be downloaded from https://github.com/max-ionov/rucoref/tree/master/notebooks/coreference-dialog-2017

2. *Head match*: True if both NPs are not pronouns and an antecedent candidate head matches an anaphoric NP head. This feature is a simple analogue for an *Alias* feature.

3.3. Baseline experiment results, Rule-based

The first four systems that we created were rule-based. They were designed to yield very high precision sacrificing the recall:

- StrMatch: two NPs corefer if their lemmas are the same (only for nouns and deictic pronouns).
- StrMatchPro: like the previous one, only non-deictic pronouns are paired with the nearest NP that agrees in gender and number.
- HeadMatch: two NPs corefer if their heads are the same (only for nouns and deictic pronouns).
- HeadMatchPro: like the previous one, only non-deictic pronouns are paired with the nearest NP that agrees in gender and number.

The results for the baseline systems are given in the tables 1 and 2.

Table 1: Rule-based coreference systems, gold mentions

	MUC			**B^3**		
	P	**R**	**F$_1$**	**P**	**R**	**F$_1$**
StrMatch	94.29	37.36	53.52	97.09	38.19	54.82
StrMatchPro	84.90	52.42	64.82	89.34	43.35	58.37
HeadMatch	87.78	47.06	61.27	92.11	43.64	59.22
HeadMatchPro	84.89	52.50	64.87	89.29	43.38	58.40

While the HeadMatch and the StrMatch baselines resolvers show very high precision, two other algorithms increase the recall by adding resolving personal pronouns. Even though the precision is much lower in the latter two cases, the overall quality is still better.

Nevertheless, the application of these algorithms are obviously very limited.

Table 2: Rule-based coreference systems, gold boundaries, mention detection f-score 51.38

	MUC			**B^3**		
	P	**R**	**F$_1$**	**P**	**R**	**F$_1$**
StrMatch	52.86	32.29	40.09	33.54	34.04	33.79
StrMatchPro	34.40	45.46	39.16	26.89	39.58	32.02
HeadMatch	35.26	41.38	38.07	29.57	38.88	33.59
HeadMatchPro	34.40	45.49	39.18	26.89	39.58	32.02

3.4. Baseline experiment results, ML approach

To incorporate more features without a need of combining them and handcrafting a set of rules we created a coreference resolution system based on a classic system by Soon et al. ([16]).

We used a decision tree classifier implemented in the scikit-learn Python module ([12]). Training and test instances were generated in the following way: for each anaphor-antecedent pair one positive example was generated. Also, a negative example was created for each candidate antecedent between the true pair of an anaphor and an antecedent.

For a baseline ML classifier we used a set of 11 features for the classifier:

1. The distance between an anaphoric NP and a candidate antecedent is 1 sentence.
2. Both NPs are not pronouns and after removing any demonstratives they match.
3. NPs agree in animacy and if they are not pronouns their syntactic heads match.
4. Anaphoric NP is a pronoun.
5. Candidate antecedent is a pronoun.
6. Both NPs are pronouns.
7. NPs agree in gender.
8. NPs agree in number.
9. Both NPs are proper.
10. An anaphoric NP is a demonstrative.
11. NPs are in the appositive relation.

Most of the features were taken from the original paper, some other were adapted to use with Russian (see 3.2 for details).

In order to decrease the noise in the data, we tweak the classifier setting the minimum number of samples required to be at a leaf node to 1% of the training samples. This simplifies the tree and makes the classifier less prone to overfitting.

The results are presented in the MLMentionPair row in the table 3 for the gold mentions case and 4 for the predicted mentions case. The classifiers show slightly better results than the rule-based ones: lowering the precision, it increases the recall.

Interestingly, training the classifier with a feature *Head Match* without any restrictions on part of speech yields better results which asymptotically approach the results of *HeadMatch* baseline classifier. A further analysis of feature importances for the classifier shows that this feature is the only one that takes part in classification decisions in this case.

To improve the baseline results, we added more features to the classifiers that can be grouped into 4 classes: *distance* features, *morphological*, *lexical* and *syntactical*.

The *Distance* group we contains the original distance feature and the binary feature if there are more than 3 nouns between the NPs. Other distance features that were tested (either in terms of nouns, NPs or words did not lead to an increase in quality).

The *Morphological* group consists of binary features checking if NPs are the pronouns of a specific type: deictic, relative, reflexive or possessive. This group increased the quality of noun-pronoun coreference resolution drastically.

Lexical features are two heuristics: a feature showing if one of the NPs equals to a noun modifier in another NP. This feature allows to resolve the cases like *prezident Obama* 'president Obama'—*prezident* 'the president' even if the head of the first NP is *Obama*. The second feature is a simple heuristic for acronym detection.

Syntactic features checks if either of NPs is a subject, an object, whether they are situated in the beginning of a sentence and whether they are both subjects (syntactic parallelism).

Table 3: ML-based coreference systems, gold mentions

	MUC			**B^3**		
	P	**R**	**F$_1$**	**P**	**R**	**F$_1$**
HeadMatchPro	84.89	52.50	64.87	89.29	43.38	58.40
MLMentionPair	73.98	62.24	67.61	71.40	49.34	58.36
MLUpdated	79.29	63.01	70.25	79.42	48.39	60.14

Table 4: ML-based coreference systems, gold boundaries, mention detection f-score 51.21

	MUC			**B^3**		
	P	**R**	**F$_1$**	**P**	**R**	**F$_1$**
HeadMatchPro	34.40	45.49	39.18	26.89	39.58	32.02
MLMentionPair	37.91	55.85	45.16	21.88	43.98	29.22
MLUpdated	37.94	53.87	44.52	25.00	42.61	31.51

The classifier with a full feature set outperforms both the naive baseline and the baseline ML system on gold mentions but does not performs so well on predicted mentions. The B^3 metric shows that the MLUpdated classifier is more precise than the basic ML classifier but less precise than the rule-based baseline, whereas it shows more recall than the rule-based classifier and slightly more than the basic ML one. The reason behind this is that it handles more cases than both classifiers and it can correctly distinguish more cases than the basic ML system.

Still, it is impossible to resolve correctly coreferent NPs which are, for example, synonyms. To do so, we need to incorporate semantic information into the classifier feature set.

3.5. Incorporating semantic information

Incorporation of some semantic information has shown to be very useful for coreference resolution. Named entity detection can improve the quality by giving the possibility to compare semantic classes of two mentions. Named entity linking can resolve the coreference, stating that two NE should be linked to the same object. Measuring semantic relatedness between two NPs, we can get the probability of two mentions to be coreferent.

As it was already mentioned, there is a limited amount of NLP tools and resources that work with Russian that are available for research purposes. For example, there is no publicly available NER detector. There is a freely available corpus with annotated named entities that can be used for train the NER detector[4], but creating a named entity resolution system is out of the scope of this paper. Overall situation with resources that can be used to extract semantic information is becoming much better over the last few years: A subset of the RuThes ontology was made available as RuThes-Lite ([9]), YARN, a crowdsourcing project to build a WordNet for Russian is developing rapidly ([3]), word2vec models trained on Russian texts are available as a part of the project "RusVectōrēs" ([8]).

In this paper we compare 3 different instruments to integrate the semantic information in the coreference resolution system:

1. A list of named entities with their types and possible synonyms.
2. A word2vec model to check if the two NPs are synonyms.
3. A thesaurus to check if two NPs are synonyms or one of them is a more general term for another.

With the aid of those instruments we can improve both an *Alias* and a *Semantic agreement* features in our classifier:

1. One NP is an alias of another.
2. NPs agree in animacy and if they are not pronouns they are semantically compatible if there is information about the semantic class of the mentions. Otherwise their heads should match.

For a first experiment, we constructed two lists of named entities. First, we compiled a small list which contained 5 frequent NEs from the training set. In the second iteration we used the GeoNames database[5] to create a list of geographical names as named entities. In total 32 934 names were used. Both experiments showed an improvement over a baseline system. Even using the small list improved the recall for the coreference resolution, improving the F-measure as a result. Further extension of the list improves the results further.

For the second experiment we employed the "RusVectōrēs" word2vec model. As a preliminary experiment we used it only to enrich the *Alias* feature. If the semantic similarity between the heads of the two NPs were more than the threshold, the NPs were considered aliases. This approach gave a slight improvement over the initial results, improving the recall and decreasing the precision. The reason behind the small impact is that there were very few cases when the similarity between NPs were big enough. Still, as described below in 4, this method allows to join the NPs that cannot be joined without the semantic information. There are other possible ways to employ word2vec models, but they are not covered in this paper and are for future research.

The third source of semantic information was RuThes-Lite, a thesaurus with several relations between concepts and a set of string representations for each concept. We used it to implement an *Alias* feature and to replace the *Semantic agreement* feature: if the heads of two NPs are the synonyms in RuThes-Lite, or there is a path

[4] https://github.com/dialogue-evaluation/factRuEval-2016

[5] http://geonames.org/

between the heads of two NPs using the parent concept relation ('ВЫШЕ') they are considered aliases. If the domains of the heads of two NPs are the same they are considered semantically related. As in the previous case, the approach shows a slight improvement, increasing the recall of the system.

The results for all the experiments are given in Table 3 for gold mentions and Table 4 for predicted mentions.

Table 5: The impact of semantic information, gold mentions

	MUC			B^3		
	P	R	F_1	P	R	F_1
MLMentionPair	73.98	62.24	67.61	71.40	**49.34**	58.36
MLUpdated	79.35	63.44	70.51	**79.37**	48.60	60.29
NamedEntities	**79.43**	63.72	70.71	**79.37**	48.86	**60.48**
Word2vec	79.29	63.49	70.52	79.25	48.64	60.28
RuThes	79.19	63.79	70.66	78.92	48.78	60.29
All	79.19	**63.97**	**70.77**	78.85	48.94	60.39

Table 6: The impact of semantic information, gold boundaries, mention detection f-score 51.21

	MUC			B^3		
	P	R	F_1	P	R	F_1
MLMentionPair	37.91	55.85	**45.16**	21.88	**43.98**	29.22
MLUpdated	37.94	53.87	44.52	**25.00**	42.61	31.51
NamedEntities	**38.01**	54.10	44.65	24.99	42.83	**31.56**
Word2vec	37.69	53.92	44.37	24.95	42.68	31.49
RuThes	36.27	54.20	43.46	24.63	42.83	31.28
All	36.08	**54.32**	43.36	24.60	42.94	31.28

4. Discussion

In this paper we described our experiments on building a coreference resolution system for Russian. We established a baseline for Russian by building an ML-based system using the features proposed in [16], and showed that by adding shallow non-semantic features we can improve its F-measure by 2–3%.

Our experiments with adding semantic information from various sources showed that even the tiniest bits of semantic information can improve the overall quality of the system. It helps coreference linking improving the overall recall, although it usually decreases the precision. At the same time we showed that using the ontology and distributional model had a very small impact on the results.

Named entity list showed the largest impact on the results, mainly because its decrease in precision was minimal due to its nature. The main limitation of this approach is, obviously, a limited size of such list: with its growth the precision should drop due to inevitable cases of homonymy.

In the case of the distributional model, the main reason of its small impact was the small amount of cases where the heads of NPs had a similarity score higher than the threshold. With a decreased threshold there were more cases but more unwanted results (mainly co-hyponyms like *muzh* 'husband'—*zhena* 'wife'). Nevertheless, this model improved the results in some cases that were impossible without it, e.g. *muzh* 'husband'—*suprug* 'spouse'. Those cases can be easily solved also by ontologies like RuThes, as we will see below, but theoretically a distributional model trained on different kinds of texts should work well with non-standard vocabulary. Another space for an improvement in this area is to use a distributional model in a more elaborated way, not only as a filter with a threshold. This is a direction for a future research.

The impact of using the RuThes ontology was also low, again, mainly because of a small amount of cases in which it was used, but as with the distributional model, its use was crucial for some cases, e.g. *rabota* 'job'—*trud* 'labour'. The main problem of this approach was homonymy. In cases like *litso* 'face' / 'person'—*chelovek* 'person', NPs were erroneously considered as aliases. Since this is not the problem of an ontology but its usage, this can be improved in the future.

There are still cases which require semantic information which cannot be linked with the methods described in the paper. There are two important classes of them. The first one is when NPs are from the same base class but the connection between them is not universal but arises in the text. E.g. *tjotushka* 'aunt'—*pomesh'itsa* 'landlady'. In this example, there is a person who is an aunt and a landlady at the same time. This problem in principle can be solved with an ontology but this solution may increase the noise in the output.

Another problem arises when the equality between the NPs are derived from the world knowledge. Like *ministr* 'minister'—*pomosh'nik prezidenta* 'a person who helps the president'. This kind of information cannot be extracted from the ontology (at least, not all the cases, even if an ontology contains some specific relations to tackle this problem).

References

1. *Bagga A., Baldwin B.* (1998), Algorithms for scoring coreference chains, The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, pp. 563–566.
2. *Bobrow D. G.* (1964), A question-answering system for high school algebra word problems, Proceedings of the October 27–29, 1964, fall joint computer conference, part I, ACM, pp. 591–614.
3. *Braslavski P., Ustalov D., Mukhin M., and Kiselev Y.* (2016), Yarn: Spinning-in-progress, Proceedings of the Eight Global Word net Conference, pp. 58–65.
4. *De Marneffe M.-C., Recasens M., C. Potts* (2015), Modeling the lifespan of discourse entities with application to coreference resolution, Journal of Artificial Intelligence Research, vol. 52, no. 1, pp. 445–475.
5. *Hobbs J.* (1978), Resolving pronoun references, Lingua, Vol. 44, pp. 311–338.
6. *Ionov M., Kutuzov A.* (2014), Influence of morphology processing quality on automated anaphora resolution for Russian, Proceedings of the international conference "Dialogue-2014", Moscow.

7. *Ionov M., Toldova S.* (2016), Identification of singleton mentions in Russian, CEUR Workshop Proceedings, in press.
8. *Kutuzov A., Andreev I.* (2015), Texts in, meaning out: neural language models in semantic similarity task for Russian, Proceedings of the International Conference on Computational Linguistics and Intellectual Technologies "Dialogue 2015", Moscow, Vol. 2, pp. 133–145.
9. *Loukachevitch N., Dobrov B., Chetviorkin I.* (2014), Ruthes-lite, a publicly available version of thesaurus of Russian language Ruthes, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference Dialogue, Bekasovo, Russia, pp. 340–349.
10. *Ng V., Cardie C.* (2002), Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution, Proceedings of the 19th International Conference on Computational Linguistics, Volume 1, ser. COLING'02. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1–7.
11. *Novák M., Žabokrtský Z.* (2011), Resolving noun phrase coreference in Czech, 8th Discourse Anaphora and Anaphor Resolution Colloquium, DAARC 2011, Springer Berlin Heidelberg, pp. 24–34.
12. *Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E.* (2011), Scikit-learn: Machine learning in Python, Journal of Machine Learning Research, vol. 12, pp. 2825–2830.
13. *Pradhan S., Luo X., Recasens M., Hovy E., Ng V., Strube M.* (2014) Scoring coreference partitions of predicted mentions: A reference implementation, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 2: Short Papers, Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 30–35. [Online]. Available at: http://www.aclweb.org/anthology/P14-2006
14. *Savary A., Ogrodniczuk M., Zawislawska M., Glowinska K., Kopec M.* (2015), Coreference: Annotation, Resolution and Evaluation in Polish, Walter de Gruyter GmbH, Berlin.
15. *Sharoff S., Nivre J.* (2011), The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge, Proceedings of the International Conference on Computational Linguistics and Intellectual Technologies "Dialogue 2011", Bekasovo, pp. 591–605.
16. *Soon W. M., Ng H. T., Lim D. C. Y.* (2001), A machine learning approach to coreference resolution of noun phrases, Computational linguistics, vol. 27, no. 4, pp. 521–544.
17. *Soraluze A., Arregi O., Arregi X., Ceberio K., De Ilarraza A. D.* (2012), Mention detection: First steps in the development of a basque coreference resolution system, Proceedings of KONVENS 2012, pp. 128–136.
18. *Toldova S., Grishina Y., Ladygina A., Sim G., Kurzukov M., Azerkovich I., Vasilyeva M.* (2014), Coreference corpus in Russian, Program & Book of Abstracts. CILC 2014. Las Palmas de Gran Canaria, Aelinco, pp. 154–155.
19. *Vilain M., Burger J., Aberdeen J., Connolly D., Hirschman L.* (1995), A model-theoretic coreference scoring scheme, Proceedings of the 6th Conference on Message Understanding, ser. MUC6 '95. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995, pp. 45–52. [Online], available at: http://dx.doi.org/10.3115/1072399.1072405

A SYNTAX-BASED DISTRIBUTIONAL MODEL FOR DISCRIMINATING BETWEEN SEMANTIC SIMILARITY AND ASSOCIATION

Trofimov I. V. (itrofimov@gmail.com),
Suleymanova E. A. (yes2helen@gmail.com)

Program Systems Institute of RAS, Pereslavl-Zalessky, Russia

In recent years, distributional semantics has shown a trend towards a deeper understanding of what semantic relatedness is and what it is composed of. This is attested, in particular, by the emergence of new gold standards like SimLex999, WS-Sim and WS-Rel. Evidence from cognitive psychology suggests that humans distinguish between two basic types of semantic relations: category-based similarity and thematic association. The paper presents a distributional model capable of differentiating between these relations, and a dataset consisting of 500 similar and 500 associated pairs of nouns that can be used for evaluation of such models.

Keywords: semantic relatedness, semantic similarity, association, RUSSE, RuSim1000 dataset, syntactic-based distributive semantic model, RuWac, MaltParser

ДИСТРИБУТИВНАЯ МОДЕЛЬ ДЛЯ РАЗЛИЧЕНИЯ СЕМАНТИЧЕСКОГО СХОДСТВА И АССОЦИАЦИИ

Трофимов И. В. (itrofimov@gmail.com),
Сулейманова Е. А. (yes2helen@gmail.com)

Институт программных систем РАН,
Переславль-Залесский, Россия

1. Introduction

Measuring semantic relatedness of words or concepts plays an important role in the tasks of text categorization, search query expansion and many others. Of particular interest is a more specific case of semantic relatedness—semantic similarity, reflecting categorical commonality of terms (concepts). Semantic similarity has its special applications, for instance, in the construction of thesauri and ontologies. This article is devoted to the methods of distributional modeling that can tell semantically similar words from otherwise related cases. The models are designed for differentiating pairs of similar Russian-language nouns from those of thematically related ones, based on their syntactic context. This research complements the state of the art presented during RUSSE—the First Workshop on Russian Semantic Similarity Evaluation [25].

In 2015, the first RUSSE Workshop performed a systematic comparison and evaluation of different approaches to developing distributional semantic models aimed at revealing and measuring the degree of semantic similarity[1] of terms. Distributional models of semantics encode meanings of words as vectors in a highly dimensional space of context words. Similarity of word meanings is then measured as similarity of vectors. Such context vectors can be formed in a multitude of ways. The Workshop revealed that, for Russian, skip-gram [22] models currently perform the best, although other distributional approaches are only slightly behind:

- a classical DSM [30], where vectors are composed of most frequent Russian nouns, verbs, adjectives and adverbs;
- the GloVe model [27]; its application to Russian is described in [19];
- the CBOW [22]; experiments with the model are reported in [16].

Our research is an attempt to develop distributional models aimed at differentiating between two kinds of semantic relations:

- relations that are based on shared intrinsic features and common category membership (similarity);
- relations that stem from thematic, or situational, co-occurrence and are not supported by taxonomical commonality (associations).

Associations are given lower weights by our models. For context vector composition, we use a selective syntactic dependency approach: we only include the words that have a specific dependency relationship with the target word. Our measures are of the global type, as opposed to contextual ones [3], in that we do not use any context for meaning disambiguation. For evaluation, in the absence of Russian-language gold standards for testing the ability of the model to discriminate similarity from association, we compiled our own dataset RuSim1000.

2. Terminological issues

The notion of similarity is central to the domain of distributional semantics. In [31], the general idea behind the famous Distributional Hypothesis—a set of statements attributed to different authors[2]—is summed up as follows: "[...] there is a correlation between distributional similarity and meaning similarity, which allows us to utilize the former in order to estimate the latter".

Whereas the former—distributional similarity—can have a clearly defined mathematical sense, there is no common understanding of what the latter is.

[1] We retain the term 'similarity' here, as it was used by the Workshop organizers; the right term would be 'relatedness' (terminological issues will be discussed in the next section).

[2] The most cited version seems to be that of Rubenstein and Goodenough, 1965 (see reference item [29]): "words which are similar in meaning occur in similar contexts".

The intuitive notion of similarity has proved very hard to define precisely. Even psychology, where similarity is one of the most central theoretical constructs, has not come up with a commonly agreed definition.

Similarity in its broad sense is very flexible. Many seem to agree that similarity of two things can only be judged with respect to some X. These ‘respects’ for similarity are determined by factors that are intrinsic to the comparison process [21]. As a consequence, two concepts (or two words) are not intrinsically similar or dissimilar [4]. The relative importance of common and different features depends on the task or context (solving odd-one-out puzzles is an illustrative example).

Before we address similarity issues from the prospective of distributional semantics, it is worth noting that, as evidenced by cognitive research [18], humans have distinct neural systems for two types of knowledge: feature-based taxonomic (categorical) knowledge and thematic knowledge—the “grouping of concepts by participation in the same scenario or event” [23].

In line with this distinction, two basic types of conceptual relations are distinguished. The first is feature-based taxonomic relatedness—this relation is commonly referred to as ***(semantic) similarity*** [10], [13], [1]. Semantic similarity refers to sharing of ‘intrinsic’ (perceptual or functional) features that account for membership in the same semantic category. *Car* and *bike* are said to be semantically similar because of their common physical features (wheels), their common function (transport), or because they fall within a clearly definable category (modes of transport) (example taken from [13]). Other terms for semantic similarity are *semantic category relatedness* [10], *taxonomic similarity / taxonomic relatedness* [17].

In contrast, the second type of relation—thematic relatedness—is based on co-occurrence (linguistic or otherwise) or functional relationships. Entities represented by thematically related concepts frequently occur together “in space and language” [13]. While similarity is based on feature overlap, thematically related concepts (entities) are not supposed to share intrinsic properties, although this is not impossible. In the psychological literature, thematic relatedness is mainly referred to as ***association*** [10], [13], although it is also known as *thematic similarity* [34], *topical similarity* [12], *domain similarity* [33], *thematic relatedness* [17], *relatedness* (as used in [1], but not in [7] and [8]—see below). Association is exemplified by pairs *car-petrol*, *bee-honey*.

Similarity and association are two distinct relations, “neither mutually exclusive nor independent” [13]. Two related concepts can be

(1) similar and associated (*coffee-tea, brandy-wine, king-queen, doctor-nurse, blouse-skirt*);
(2) associated, but not similar (*coffee-mug, king-crown, engine-car, cow-milk*);
(3) similar, but not associated (*bear-cow, house-cabin, nurse-lawyer*).

2.1. Similarity versus association in NLP

With few exceptions [33], recent research in distributional semantics appears to have been focused on quantitative rather than qualitative aspects of word interaction within lexical semantic system. Such approaches neglect the difference between similarity and association [28], [14], [20], as their focus is estimating the strength of the connection between two words in the semantic network, regardless of the relation type. Such connection is most often referred to as ***relatedness*** [7], [8], [26] in the broad sense. Thus understood, semantic relatedness subsumes both semantic similarity and thematic association as its specific cases.

Until recently, there has been some confusion in terminology regarding the object of distributional semantic modeling within this paradigm. What is referred to as measuring 'similarity' as conveyed by distributional similarity turns out to be in fact estimating relatedness. Thus, the term 'semantic similarity' is sometimes taken to be the synonym for semantic relatedness and semantic proximity, and the inverse of semantic distance. For the sake of justice, it should be noted that in recent publications such terminological ambiguities are becoming rare.

Most of the gold standard datasets designed for the evaluation of distributional semantic models do not distinguish between taxonomic, feature-based similarity and thematic association (WordSim-353 [11], MEN Test Collection [6], RUSSE HJ [25]).

The utility of such resources to the development and application of distributional models is limited. This is still more so, because "many researchers appear unaware of what their evaluation resources actually measure" [13].

Recently developed resources, like SimLex999 [13], WS-Sim and WS-Rel subsets of WordSim353 [1], are expected to fill this gap. It is questionable, though, to what extent these resources can serve to actually measure the ability of models to reflect similarity as opposed to association. The developers of WS-Sim, for example, turned to human subjects to separate between similar and (otherwise) related cases, but left the original WordSim353 scores intact.

3. RuSim1000 dataset

The dataset was developed with the aim of evaluating Russian-language distributional models that focus on revealing *similarity* (possibly accompanied by association) as opposed to pure *association*.

Similarity, or taxonomical, feature-based similarity,—semantic relation that is based on shared intrinsic features and common category membership.

Association—semantic relation that stems from thematic, or situational, co-occurrence and is not supported by taxonomical (ontological) commonality.

RuSim1000 is composed of 1000 pairs of *related* nouns that are divided into two subsets—the sets of positive and negative examples. Positive examples are pairs of similar nouns. Negative examples are pairs of associated, but not similar nouns. Pairs of similar words that are also associated (*король-королева*, *king-queen*) are positive examples: it is the presence or absence of similarity that matters.

The core of the positive subset is formed by the following cases:

- synonyms (*имя-название, name-title*) and near synonyms (*особенность-аспект*, *peculiarity-aspect*);
- hyponym-hypernym (*имя-прозвище, name-nickname*) and the inverse (*питон-змея, python-snake*);
- co-hyponyms (*писатель-поэт, writer-poet*).

Clear-cut negative cases are pairs of nouns representing ontologically different entities linked by any of the following relations:

- part-whole (*шерсть-животное, fur-animal*) and the inverse (*лошадь-грива, horse-mane*);
- element-set (*самолет-эскадрилья, airplane-squadron*) and the inverse;
- functional (situational) relationship (*доктор-клиника, doctor-clinic, винтовка-выстрел, rifle-shot*);
- free association (*край-земля, edge-land*).

For a number of difficult and borderline cases the following decisions were made:

- Antonyms. Contrary to the intuition that antonymous terms are dissimilar, we take them to be similar (i.e. positive examples)—due to an assumption that their opposition is likely to hold within a certain category, to which they both belong (*свет-тьма, light-darkness*).
- Roles. As long as the taxonomy of roles should be separate from the taxonomy of types (or, at least, the conceptual difference between types and roles should be taken into account by the ontology), the category-membership criterion is somewhat difficult to apply to those pairs that are a mixture of a type and its role. It was decided to qualify as positive (i.e. similar):
  - pairs of the kind “a type and its typical role” (*торф-топливо, peat-fuel*, but not *самолет-вооружение, airplane-armament*);
  - thematically related roles of the same holder type, including complementary roles (*врач-медсестра, doctor-nurse, врач-пациент, doctor-patient*).

Table 1. Positive (similar words) and negative (associated, but not similar words) examples from RuSim1000

word1	**word2**	**sim**
лошадь (horse)	жеребец (stallion)	1
лошадь (horse)	кобыла (mare)	1
лошадь (horse)	пони (pony)	1
лошадь (horse)	кляча (jade)	1
лошадь (horse)	седло (saddle)	0
лошадь (horse)	конюх (groom)	0
лошадь (horse)	грива (mane)	0
лошадь (horse)	галоп (gallop)	0

RuSim1000 was designed in such a way that it would be compatible with the RUSSE evaluation framework[3]. Average Precision (AP) [35] used by RUSSE RT was chosen as evaluation measure. AP is calculated for a ranked list of examples. The higher the rank of the positive examples, the more they contribute to the AP (see formula 1).

$$Average\ Precision = \frac{\sum_r P@r}{R} \tag{1}$$

[3] http://russe.nlpub.ru/downloads/

where *r* is the rank of each positive example, *R* is the total number of positive examples, *P@r* is the precision of the top-r examples.

Positive and negative examples in RuSim1000 are equal in number. As a consequence, the random baseline is about 0.5 [5]. It is carefully observed that there is equal number of positive and negative pairs beginning with the same word. Thus we could evaluate our algorithms with the same evaluation tools as were used for RUSSE RT.

The dataset is available at: https://zenodo.org/record/546238#.WPDyi6IlGUk

4. Semantic similarity measures

The way objects, events, phenomena etc. are categorized by humans is very much dependent on their activities. Despite rather flexible and 'non-systemic' character of categorization, which makes formal definition of shared categorical membership in terms of feature overlap almost impossible, concepts in many categories do share perceptual and functional features. The intuition behind our distributional models of similarity is as follows:

- similar objects tend to have more shared features than dissimilar;
- similar objects tend to act in similar way;
- similar objects tend to be exposed to similar actions.

That means we expect our similarity measures to have a clear interpretation as similarity of features and behavior.

Put in linguistic terms, the three above statements (roughly) read as follows:

- similarity of features is evidenced as sharing common adjectives;
- similarity of behavior manifests itself as being the subject and/or the object of the same verbs.

Three types of syntactic relations (as defined in SynTagRus [2]) are retrieved from the source corpus:

- attributive (for feature-based similarity);
- predicative and 1-completive (for behavioral similarity).

The context vector is composed of adjectives, for feature-based similarity measure, and of verbs—for behavioral similarity. The length of vectors is not limited.

The idea of using syntactic features for distributional modeling is not new. There is a comprehensive survey on the topic in [24]. The authors suggest an elegant general framework for the integration of syntax-based and word co-occurrence approaches. A syntax-based approach to vector formation (for Russian word categorization) is presented in [15]. There is a hypothesis that models that learn from input annotated for syntactic or dependency relations better reflect similarity, whereas approaches that learn from running-text or bag-of-words input better model association [13]. However, we do not know of any attempt of applying syntax-based approaches for discriminating between similarity and association.

Back to our model, two questions are to be answered:

- how to form context vectors out of absolute frequencies of the target syntactic relations;
- how to measure the distance between vectors.

As far as English is concerned, Bullinaria and Levi [9] showed that the following combination proved to be working well for semantic relatedness evaluation: the positive pointwise mutual information (formula 2) as vector component value and cosine similarity for measuring the distance between vectors.

Let
t be the target noun,
c—a context word (a component of a context vector).
Then the pointwise mutual information *pmi(c, t)* is calculated as:

$$pmi(c,t) = log_2 \frac{p(c,t)}{p(c) \cdot p(t)},$$

where *p(c,t)* is the probability that *t* and *c* occur linked by the target relation,
p(c) and *p(t)* are the probabilities of independent occurrence of *c* and *t*, respectively.

The positive pointwise mutual information *ppmi(c, t)*:

$$ppmi(c,t) = \begin{cases} pmi(c,t), if\ pmi(c,t) \geq 0; \\ 0, \quad\quad if\ pmi(c,t) < 0. \end{cases} \tag{2}$$

We used ppmi with a single reservation: the probabilities *p(c)* and *p(t)* were calculated on the set of relations of the given type rather than on the entire corpus. To put it otherwise, for feature-based measure, for example, *p(c)* и *p(t)* are calculated on the set of noun-adjective pairs extracted from the corpus.

Besides, the cosine distance was taken to be zero if there were less than 10 non-zero summands in the numerator (formula 3).

$$\cos(A,B) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\|A\| \cdot \|B\|} \tag{3}$$

5. Experiments and results

As a source of statistical information about syntactic relations we used the RuWac[4] corpus that had been syntactically annotated with MaltParser[5] [32]. A total of about 223 million target relation instances were extracted (Table 2).

Table 2. The statistical data obtained from RUWAC (the figures are given in million pairs)

	attributive	predicative	1-completive	total
number of relation instances	114	51	57	223
number of unique relations (lexeme$_1$, lexeme$_2$, relation_ type)	11.1	9.2	8.7	29

[4] http://corpus.leeds.ac.uk/tools/ru/ruwac-parsed.out.xz (as of December 2016)

[5] http://corpus.leeds.ac.uk/mocky/

The data was used to develop distributional models that were evaluated against the RuSim1000 dataset.

The results of the tests[6] are presented in Tables 3 and 4. The Table 3 lists the average precision scores for the similarity/association discrimination task obtained by three models that learnt from single-syntactic-relation annotated input. Table 4 shows the average precision of the same categorization yielded by the models that learnt from combinations of two syntactic relations.

Table 3. Testing results for single-relation input

syntactic relation		
attributive	**predicative**	**1-completive**
0.907	0.846	0.882

Table 4. Testing results for combined input

combination of syntactic relations	
attributive + predicative	**attributive + 1-completive**
0.918	0.925

The experiments confirmed that a rather limited one- or two-relation syntactic context is sufficient to discriminate between similar and associated cases (this task being quite different from that of similarity or association scoring).

6. Conclusions

The paper focuses on the two cognitive and linguistic phenomena that account for semantic relatedness of terms—those of taxonomic similarity and thematic association. Dataset RuSim1000 is presented—a gold standard that can be used to evaluate the ability of models to discriminate between the two types of conceptual relations. Distributional models for similarity/association discrimination were developed, in which syntactic features of terms were used as 'proxies' for feature-based and behavioral similarity of objects. The experiments proved that the models are good enough at the task in hand (average 0.9 on RuSim1000).

The research was carried out as part of the project "Methods for automated extraction of events from texts" (No. AAAA-A17-117040610371-7).

[6] The tests were performed with the software used for RUSSE—http://russe.nlpub.ru/downloads/

References

1. *Agirre E., Alfonseca E., Hall K., Kravalova J., Pasca M., and Soroa A.* (2009), A study on similarity and relatedness using distributional and Wordnet-based approaches, Proceedings of NAACL, Boulder, pp. 19–27.
2. *Apresian Ju. D., Boguslavsky I. M., Iomdin B. L. et al.* (2005), Syntactically and Semantically Annotated Corpus of Russian: State-of-the-Art and Prospects [Sintaksicheski i semanticheski annotirovannyj korpus russkogo jazyka (sovremennoe sostojanie i perspektivy], National Corpus of Russian 2003–2005 (Results and Prospects) [Natsionalnyj korpus russkogo jazyka 2003–2005 g. (rezul'taty i perspektivy)], Moscow, Indrik, pp. 193–214.
3. *Arefyev N. V., Panchenko A. I., Lukanin A. V., Lesota O. O., Romanov P. V.* (2015), Evaluating three corpus-based semantic similarity systems for russian, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", RGGU, pp. 106–118.
4. *Batchkarov M., Kober T., Reffin J., Weeds J., Weir D.* (2016), A critique of word similarity as a method for evaluating distributional semantic models, Proceedings of The First Workshop on Evaluating Vector Space Representations for NLP, Berlin, pp. 7–12.
5. *Bestgen Y.* (2015), Exact Expected Average Precision of the Random Baseline for System Evaluation, The Prague Bulletin of Mathematical Linguistics, 103, pp. 131–138.
6. *Bruni E., Tran N. Kh., Baroni M.* (2014), Multimodal Distributional Semantics, Journal of Artificial Intelligence Research (JAIR), 49, pp. 1–47.
7. *Budanitsky A., Hirst G.* (2001), Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, pp. 29–34.
8. *Budanitsky A., Hirst G.* (2006), Evaluating Wordnet-based measures of lexical semantic relatedness, Computational Linguistics, 32(1), pp. 13–47.
9. *Bullinaria J., Levy J.* (2007), Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study, Behavior Research Methods, 39(3), pp. 510–526.
10. *Chiarello C., Burgess C., Richards L., Pollock A.* (1990), Semantic and associative priming in the cerebral hemispheres: some words do, some words don't ... sometimes, some places, Brain and Language, 38, pp. 75–104.
11. *Finkelstein L., Gabrilovich E., Matias Y., Rivlin E., Solan Z., Wolfman G., Ruppin E.* (2001), Placing search in context: The concept revisited, Proceedings of the Tenth International World Wide Web Conference, Hong Kong, pp. 406–414.
12. *Hatzivassiloglou V., Klavans J. L., Holcombe M. L., Barzilay R., Kan M.-Y., McKeown K.* (2001), Simfinder: A flexible clustering tool for summarization, Proceedings of the NAACL Workshop on Automatic Summarization, Pittsburgh, available at: http://hdl.handle.net/10022/AC:P:20214.
13. *Hill F., Reichart R., Korhonen A.* (2015), SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation, Computational Linguistics 41(4), pp. 665–695.

14. *Huang E. H., Socher R., Manning C. D., Ng A. Y.* (2012), Improving word representations via global context and multiple word prototypes, Proceedings of ACL, Jeju Island, Korea, pp. 873–882.
15. *Klyshinsky E. S., Kochetkova N. A., Logacheva V. K.* (2013), Clustering words with similar sense using information about their syntactic dependencies [Metod klasterizacii slov s ispol'zovaniem informacii ob ikh sintaksicheskoj sviaznosti], Scientific and technical information. Series 2: Information processes and systems [Nauchno-tekhnicheskaja informacija. Serija 2: Informacionnye processy i sistemy], 11, pp. 36–43.
16. *Kutuzov A., Andreev I.* (2015), Texts in, Meaning out: Neural Language Models in Semantic Similarity Tasks for Russian, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", RGGU, pp. 133–144.
17. *Landrigan J.-F., Mirman D.* (2016), Taxonomic and Thematic Relatedness Ratings for 659 Word Pairs, Journal of Open Psychology Data, 4: e2.
18. *Lewis G. A., Poeppel D., Murphy G. L.* (2015), The neural bases of taxonomic and thematic conceptual relations: An MEG study, Neuropsychologia, 68, pp. 176–189.
19. *Lopukhin K. A., Lopukhina A. A., Nosyrev G. V.* (2015), The Impact of Different Vector Space Models and Supplementary Techniques on Russian Semantic Similarity, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", RGGU, pp. 145–153.
20. *Luong M.-Th., Socher R., Manning C. D.* (2013), Better word representations with recursive neural networks for morphology, Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL-2013), Sofia, pp. 104–113.
21. *Medin D. L., Goldstone R. L., Gentner D.* (1993), Respects for similarity, Psychological Review, 100(2), pp. 254–278.
22. *Mikolov T., Chen K., Corrado G., Dean J.* (2013), Efficient Estimation of Word Representations in Vector Space, arXiv preprint arXiv:1301.3781.
23. *Mirman D., Walker G. M., Graziano K. M.* (2011), A Tale of Two Semantic Systems: Taxonomic and Thematic Knowledge, Proceedings of the 33th Annual Meeting of the Cognitive Science Society (CogSci 2011), Boston, pp. 2211–2216.
24. *Pado S., Lapata M.* (2007), Dependency-based Construction of Semantic Space Models, Computational Linguistics, 33(2), pp. 161–199.
25. *Panchenko A., Loukachevitch N., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015), RUSSE: The First Workshop on Russian Semantic Similarity, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", RGGU, pp. 89–105.
26. *Panchenko A., Ustalov D., Arefyev N., Paperno D., Konstantinova N., Loukachevitch N. and Biemann C.* (2016), Human and Machine Judgements about Russian Semantic Relatedness, Proceedings of the 5th Conference on Analysis of Images, Social Networks, and Texts (AIST'2016), Communications in Computer and Information Science (CCIS), Springer-Verlag Berlin Heidelberg, pp. 221–235.

27. *Pennington J., Socher R., Manning C. D.* (2014), Glove: Global vectors for word representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.
28. *Reisinger J., Mooney R. J.* (2010), A mixture model with sharing for lexical semantics, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP), Massachusetts, pp. 1173–1182.
29. *Rubenstein H., Goodenough J. B.* (1965), Contextual correlates of synonymy, Communications of the ACM (Comp. linguistics), 8(10), pp. 627–633.
30. *Ryzhova D. A., Kyuseva M. V.* (2015), On the Nature of Semantic Similarity and It's Measuring with Distributional Semantics Models, available at: http://www.dialog-21.ru/digests/dialog2015/materials/pdf/RyzhovaDAKyusevaMV.pdf.
31. *Sahlgren M.* (2008), The Distributional Hypothesis, Rivista di Linguistica (Italian Journal of Linguistics), 20 (1), pp. 33–53.
32. *Sharoff S., Nivre J.* (2011), The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", RGGU, pp. 591–604.
33. *Turney P. D.* (2012), Domain and Function: A Dual-Space Model of Semantic Relations and Compositions, Journal of Artificial Intelligence Research (JAIR), 44, pp. 533–585.
34. *Wisniewski E. J., Bassok M.* (1996), On putting milk in coffee: the effect of thematic relations on similarity judgments, Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society, Erlbaum, pp. 464–468.
35. *Zhang E., Zhang Y.* (2009), Average Precision, Encyclopedia of Database Systems, Springer US, pp. 192–193.

EXPANDING HIERARCHICAL CONTEXTS FOR CONSTRUCTING A SEMANTIC WORD NETWORK

Ustalov D. A. (dau@imm.uran.ru)

Krasovskii Institute of Mathematics and Mechanics;
Ural Federal University, Yekaterinburg, Russia

A semantic word network is a network that represents the semantic relations between individual words or their lexical senses. This paper proposes Watlink, an unsupervised method for inducing a semantic word network (SWN) by constructing and expanding the hierarchical contexts using both the available dictionary resources and distributional semantics' methods for *is-a* relations. It has three steps: context construction, context expansion, and context disambiguation. The proposed method has been evaluated on two different datasets for the Russian language. The former is a well-known lexical ontology built by the group of expert lexicographers. The latter, LRWC ("Lexical Relations from the Wisdom of the Crowd"), is a new resource created using crowdsourcing that contains both positive and negative human judgements for subsumptions. The proposed method outperformed the other relation extraction methods on both datasets according to recall and F_1-score. Both the implementation of the Watlink method and the LRWC dataset are publicly available under libré licenses.

Keywords: lexical semantics, hyponym, hypernym, subsumption, semantic network, crowdsourcing, Russian

ПОСТРОЕНИЕ СЕМАНТИЧЕСКОЙ СЕТИ СЛОВ ПУТЁМ РАСШИРЕНИЯ ИЕРАРХИЧЕСКИХ КОНТЕКСТОВ

Усталов Д. А. (dau@imm.uran.ru)

Институт математики и механики им. Н.Н.Красовского Уральского отделения РАН; Уральский федеральный университет, Екатеринбург, Россия

Семантическая сеть слов — это сеть, представляющая семантические отношения между отдельными словами или значениями слов. В данной работе представлен метод Watlink для построения семантической сети слов на основе обучения без учителя. Метод включает три этапа: формирование иерархических контекстов, расширение иерархических контекстов, связывание полученных контекстов. Произведена оценка представленного метода на двух разных наборах данных для русского языка: по материалам тезауруса RuWordNet и по материалам нового набора данных LRWC, содержащего суждения людей о родовидовых связях русских слов, полученных при помощи краудсорсинга. Предложенный метод продемонстрировал более высокие значения полноты и F_1-меры на обоих наборах данных по сравнению с другими методами извлечения отношений. Реализация метода Watlink и набор данных LRWC доступны на условиях открытой лицензии.

Ключевые слова: лексическая семантика, гипоним, гипероним, родовидовое отношение, семантическая сеть, краудсорсинг, русский язык

1. Introduction

A semantic network is a network that represents semantic relations between concepts [27]. Such semantic networks as WordNet [8] and BabelNet [21] are successfully applied in addressing different problems requiring common sense reasoning. Construction of such a network 'by hand' is a long and expensive process that involves very large amount of efforts of expert lexicographers. For instance, the Russian language is still considered as an under-resourced natural language [12], which makes it highly topical to develop new methods for discovering and refining the available dictionaries and other lexical semantic resources in an unsupervised way.

This paper is focused on a special kind of semantic networks—semantic word networks—that represent the relations between individual words or their lexical senses rather than the entire concepts [15]. Semantic word networks (SWNs) found their application in marketing campaign optimization [25], search query expansion [11], etc. Particularly, this paper is devoted to the *hyponymy/hypernymy* relation, also known as the *is-a* or the *subsumption* relation. Thus, an SWN is a directed graph connecting the distinct lexical senses through the hypernymy relations.

The contribution of the present paper is two-fold. Firstly, Watlink, an unsupervised method for constructing an SWN that uses both distributional and dictionary resources has been proposed. Secondly, a crowdsourced dataset LRWC ("Lexical Relations from the Wisdom of the Crowd") representing both positive and negative human judgements for hyponymy and hypernymy relations for the Russian language has been disseminated. The proposed method is inspired by the one by Faralli et al. [7]. The difference is that the present method disambiguates synsets and their hierarchical contexts instead of distributional sense representations. Also, it provides an optional context expansion step to increase the lexical coverage of the resulting dataset.

The rest of the paper is organized as follows. Section 2 reviews the related work focused on the construction of *is-a* relations. Section 3 describes Watlink, an unsupervised method for semantic word network construction. Section 4 shows the evaluation results of this method on a well-known gold standard dataset for Russian. Section 5 presents the LRWC dataset and demonstrates the evaluation results on this new dataset. Section 6 concludes with the final remarks.

2. Related Work

Currently, the most widely used method for detecting hypernyms and hyponyms is the Hearst patterns [10]. These lexical-syntactic patterns, e.g., "Y such as X_1 and X_2", have successfully found a substantial number of applications including ontology learning. There is a couple of variations of such patterns like PatternSim [22] and sense definition parsing [13], but the core principle remains the same and these patterns suffer from the sparsity problem.

Various forms of crowdsourcing are used for constructing or expanding lexical resources. Wiktionary, a wiki-based dictionary, is a popular source of semantic information [31]. Also, there are other initiatives like the BabelNet Annotation Group (BANG) [21] and Yet Another RussNet (YARN) [4], which differ in the goals and deliverables.

Distributed word representations, also known as word embeddings [20], are a trending topic nowadays. Fu et al. [9] proposed the projection learning approach for constructing semantic hierarchies for the Chinese language. This approach assumes learning a linear transformation matrix such that multiplying on which a hyponym vector produces a hypernym vector. Also, the *k*-means clustering algorithm has been used to split the embeddings space into several subspaces to provide more flexibility to the model. Recently, this approach has been improved by negative sampling, which yielded a significant quality boost [28].

Shwartz et al. developed HypeNET, an integrated method that combines the syntactic parsing features with word embeddings based on a long short-term memory network [26]. However, HypeNET requires high-quality dependency pairs, which complicates its application for under-resourced languages.

3. Constructing a Semantic Word Network with Watlink

Let $\mathcal{S}$ be the input set of synsets, and a synset $S \in \mathcal{S}$ is a set of semantically equivalent word senses[1], e.g., {*auto*^{2}, *car*^{1}, *automobile*^{1}, …}. Let $\mathcal{R}$ be the input set of *is-a* relations provided in the form of tuples $(w, h) \in \mathcal{R}$, where both the hyponym w and the hypernym h have no sense labels attached, e.g., (*bank*, *building*). The goal is to assign the corresponding sense labels to these relations as well as to provide the words with missing hypernyms with those, if possible.

For that, the Watlink method, shown in Fig. 1, is proposed. Firstly, a hierarchical context representing a bag of hypernym words is constructed for each synset. Secondly, each hierarchical context is expanded using the nearest neighbor retrieval combined with projection learning. Finally, the sense labels for the hypernyms are obtained using the context disambiguation. As the result, Watlink constructs a directed graph SWN = (*V*, *E*), where $V = \bigcup_{S \in \mathcal{S}} S$ is the set of all the possible word senses appearing in all the synsets, and $E \subseteq V \times V$ is the set of disambiguated *is-a* relations between these senses.

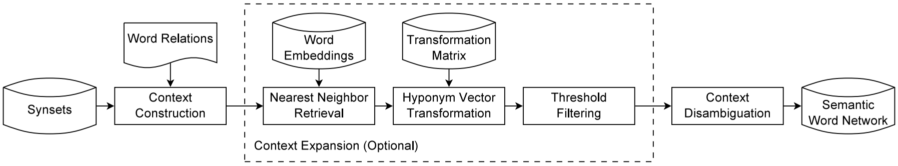


Fig. 1. An outline of the proposed method, Watlink

[1] Following the notation used in BabelNet [21], *word*^{i} denotes the *i*-th lexical sense of the given *word*.

3.1. Context Construction

A hierarchical context connects the lexical entries of the corresponding synset to their typical hypernyms in the *is-a* dictionary. For a synset $S \in \mathcal{S}$, the hierarchical context $\mathrm{hctx}(S)$ is a bag of words composed of all the hypernyms in $\mathcal{R}$ matching the words in S as hyponyms:

$$\mathrm{hctx}(S) = \{h : w \in \mathrm{words}(S), (w, h) \in \mathcal{R}\},$$

where $\mathrm{words}(S)$ is the set of lemmas corresponding to the word senses in S, (w, h) is a pair of hyponym w and hypernym h present in the dictionary $\mathcal{R}$. As the result, hypernyms are propagated to the words in the synset for which no hypernyms were available. The variable importance of words in hierarchical contexts is modeled using tf, idf, and tf-idf [19]:

$$\text{tf-idf}(h, \mathrm{hctx}(S), \mathcal{S}) = \underbrace{\frac{|h' \in \mathrm{hctx}(S) \colon h = h'|}{|\mathrm{hctx}(S)|}}_{\mathrm{tf}(h,\, \mathrm{hctx}(S))} \times \underbrace{\log\left(\frac{|\mathcal{S}|}{|S' \in \mathcal{S} \colon h \in \mathrm{hctx}(S')|}\right)}_{\mathrm{idf}(h, \mathcal{S})}$$

For example, a hierarchical context for the synset mentioned in the beginning of Section 3, can be like {*vehicle, transport, motor vehicle*} for $\mathrm{words}(S)$ = {*auto, car, automobile*}.

3.2. Context Expansion

Given the fact that the amount of available resources representing *is-a* relations is limited [12], a projection learning approach [9] has been used to expand the set of the already available subsumption pairs $(w, h) \in \mathcal{R}$. This optional context expansion step is based on searching the most similar words to h using the semantic similarity computed on word embeddings [20] and filtering out the irrelevant candidates (Fig. 2).

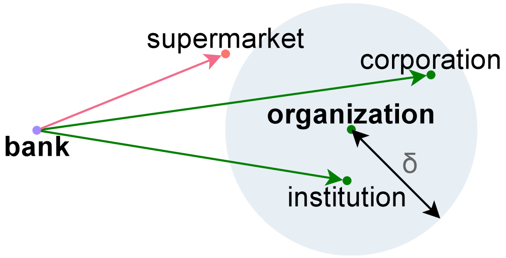


Fig. 2. Hypernyms of “bank” that are similar to a known hypernym “organization”: the candidate words “corporation” and “institution” within a radius of δ are correct, while the candidate word “supermarket” is not

Firstly, for each input *is-a* pair $(w, h) \in \mathcal{R}$, a set of n nearest neighbors $\mathrm{NN}_n(\vec{h})$ of the hypernym embedding $\vec{h}$ is retrieved. Secondly, for each hypernym candidate embedding $\vec{h'} \in \mathrm{NN}_n(\vec{h})$, a transformation matrix Φ^*, corresponding to the subspace of the vector offset $\vec{h'} - \vec{w}$, is obtained and multiplied on the hyponym embedding

$\vec{w}$ [9], resulting in the predicted hypernym vector $\Phi^*\vec{w}$. Finally, the Euclidean distance between $\Phi^*\vec{w}$ and the hypernym embedding $\vec{h}$ is computed. Those predicted vectors located within a radius of δ from the latter vector are said to be relevant: $\left\|\Phi^*\vec{w} - \vec{h}\right\| < \delta$. As the result, a hierarchical context hctx(S) can be transformed into the expanded hierarchical context $\text{hctx}'(S)$:

$$\text{hctx}'(S) = \bigcup_{\substack{w \in \text{words}(S),\\ (w,h) \in \mathcal{R}}} \{w\} \times \text{NN}^*_n\left(\vec{h}\right) \cup \text{hctx}(S)$$

where $\vec{w}$ and $\vec{h}$ are embedding vectors for the words w and h, correspondingly, $\text{NN}^*_n\left(\vec{h}\right)$ is the set of relevant candidates of n nearest neighbors of the vector $\vec{h}$.

3.3. Context Disambiguation

For each synset $S \in \mathcal{S}$ and its hierarchical context hctx(S), a sense label is estimated for each hypernym $h \in \text{hctx}(S)$. This is achieved by selecting a synset $S' \in \mathcal{S}: h \in \text{words}(S')$ that maximizes the cosine similarity [7] between hctx(S) and S to choose the optimal word sense $\hat{h}$:

$$\hat{h} = \underset{\substack{S' \in \mathcal{S}, S \neq S', h' \in S',\\ \text{words}(\{h'\}) = h}}{\arg\max} \cos(\text{hctx}(S),\ S')$$

For instance, consider the hierarchical context {*material*, *data*} and two synsets: {*material*^{1}, *textile*^{1}} and {*material*^{2}, *information*^{1}, *data*^{1}}. Using this procedure, the second sense of the word "material" will be chosen because the latter synset is more similar to the given hierarchical context. The resulting disambiguated hierarchical context contains the sense labels attached to the words composing the initial hierarchical context, i.e., $\widehat{hctx}(S) = \left\{\hat{h} : h \in \text{hctx}(S)\right\}$. It is now possible to construct an SWN with $\bigcup_{S \in \mathcal{S}} S$ as the set of nodes and $\bigcup_{S \in \mathcal{S}} S \times \widehat{hctx}(S)$ as the set of edges that are labeled *is-a* relations. An example of an SWN is presented in Fig. 3. Note that the ambiguous word "ticket" is represented twice: *ticket*^{1} is a document and *ticket*^{2} is a sign.

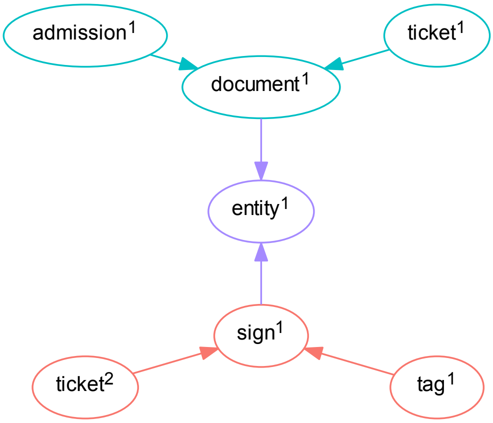


Fig. 3. An example of a semantic word network

4. Gold Standard Evaluation on RuWordNet

Since Watlink implies no strict limitations on the structure of the input synsets and *is-a* relations, it can be applied for linking senses in virtually any synset dataset, e.g., YARN [4] or UNLDC [5], with the relations from practically any subsumption dataset, e.g., Hearst patterns [10], Wiktionary [31], etc. During the gold standard evaluation, the performance of the Watlink method is studied on RuWordNet [17], a WordNet-like version of the RuThes thesaurus for Russian [16].

4.1. Experimental Setup

Given the fact that Watlink is an unsupervised method, except the optional expansion step, the synset dataset is also chosen to be obtained in an unsupervised way. A state-of-the-art method for unsupervised synset induction, Watset [29], has been used to yield the synsets from the synonymy graph composed of three synonymy dictionaries for Russian: the Russian Wiktionary [31], UNLDC [5] and the Abramov's Dictionary [1]. This resulted in 55,369 synsets uniting 83,092 lexical entries; Watset was configured to use the Chinese Whispers algorithm for word sense induction [2] and the Markov cluster algorithm for global graph clustering [30].

The following *is-a* datasets have been used in the experiments:

- *Patterns*, the dataset extracted from the lib.rus.ec electronic library using the PatternSim approach [22]; the *Limit* option specifies that only these relations appeared at least $f = 30$ times have remained.
- *Wiktionary*, the dataset extracted from the Russian Wiktionary using the JWKTL tool [31].
- *SAD* the dataset extracted from the sense definitions in the Small Academic Dictionary [6] [13].
- *Joint*, the dataset uniting *Patterns* + *Limit*, *Wiktionary* and *SAD*.

Each dataset has been expanded using the context expansion method described in Section 3.2 with $n = 10$, which is denoted as *Exp*. The transformation matrices have been estimated using projection learning with asymmetric regularization [28] on the state-of-the-art 500-dimensional skip-gram word embeddings for Russian [23]. The threshold $\delta = 0.6$ has been tuned separately on a development dataset.

4.2. Evaluation Metric

The performance is reported according to the pairwise information retrieval quality metrics: precision, recall and F_1-score [19]. For that. an *is-a* pair (*hypo*, *hyper*) is considered as predicted correctly if and only if there is a path from some sense of *hypo* to some sense of *hyper* exists in the gold standard dataset. Only the words appearing both in the gold standard and the comparable datasets are considered. The rest words are excluded from the evaluation.

4.3. Results

In most cases, the tf-idf weighing approach yielded slightly better results according to F_1-score (Fig. 4). Table 1 summarize the evaluation results obtained on the RuWordNet dataset using tf-idf weights. The highlighted results are statistically significant according to the Wilcoxon signed-rank test with the significance level of 0.01 performed similarly to [24]. It clearly seems that the expansion increases recall with a slight, yet notable, drop of precision. However, SWN outperformed the others in terms of recall and F_1-score.

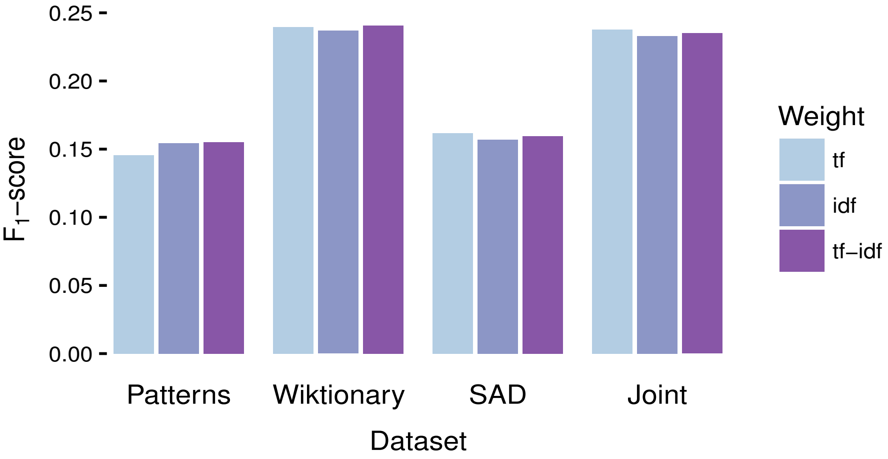


Fig. 4. Influence of the weighing approach according to the best result on each dataset

Table 1. Evaluation results on the RuWordNet dataset, only the best configurations shown, top three results are highlighted

Method	# of pairs	Precision	Recall	F_1-score
Patterns	1,597,651	0.1611	0.3255	0.2155
Patterns + SWN	236,922	0.1126	0.2451	0.1543
Patterns + Limit	10,458	**0.3773**	0.0157	0.0302
Patterns + Limit + Exp	10,715	0.3760	0.0160	0.0307
Patterns + Limit + SWN	46,758	0.1140	0.0717	0.0880
Patterns + Limit + Exp + SWN	47,387	0.1129	0.0722	0.0881
Wiktionary	108,985	**0.3877**	0.0898	0.1458
Wiktionary + Exp	110,329	**0.3874**	0.0907	0.1469
Wiktionary + SWN	177,787	0.1836	0.3460	**0.2399**
Wiktionary + Exp + SWN	179,623	0.1844	**0.3464**	**0.2407**
SAD	36,800	0.1823	0.1502	0.1647
SAD + Exp	37,702	0.1825	0.1515	0.1655
SAD + SWN	98,085	0.1383	0.1879	0.1593
SAD + Exp + SWN	99,678	0.1385	0.1883	0.1596
Joint	149,195	0.1719	0.2590	0.2067
Joint + Exp	151,150	0.1720	0.2594	0.2069
Joint + SWN	216,285	0.1685	**0.3865**	0.2347
Joint + Exp + SWN	218,290	0.1687	**0.3867**	**0.2350**

5. Lexical Relations from the Wisdom of the Crowd

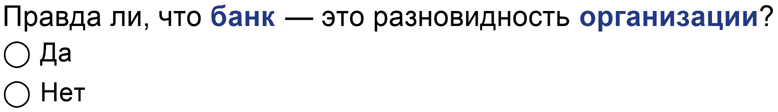
To study the performance of the proposed method more thoroughly, the best models in Section 4.3 have been chosen as the input subsumption pairs for collecting human judgements.

A set of 300 most frequent nouns have been extracted from the Russian National Corpus [18]. Then, each method or resource in Table 1, produced at most five hypernyms for each of these 300 nouns, if possible. In case it is not possible, missing answers treated as false negative answers. Two additional datasets participated in the evaluation: RuThes [16] and a noun-only version of RuWordNet [17]. The order of the extracted hypernyms is the same as in they are presented in the resource. This resulted in 10,600 unique non-empty subsumption pairs that have been passed for crowdsourcing annotation on the Yandex.Toloka[2] platform. Each pair has been annotated by seven different annotators whose mother tongue is Russian and the age is at least 20 by February 1, 2017.

Prior to this annotation, a manually composed training set of 48 tasks for less frequent nouns has been ran. Only those who answered correctly for at least 80 % of the training tasks have been permitted to complete non-training tasks for paid. Also, the workers have been provided with a detailed instruction containing recommendations among the examples of correct positive and negative answers.

5.1. Human Intelligence Task

The layout of the human intelligence task (HIT) design, depicted in Fig. 5, assumes the direct answer to a simple question: does the given pair of words represent a meaningful *is-a* relation? Since the crowd workers are not expert lexicographers and this question might be difficult for them, it has been rephrased as “Is it correct that a kitten is a kind of mammal?” (in Russian).



Правда ли, что **банк** — это разновидность **организации**?

○ Да

○ Нет

Fig. 5. Layout of the HIT: “Is it correct that a *bank* is a kind of *organization*?” (Yes / No), both the hyponym and hypernym words link to the Yandex search page; note that the hypernym is represented in the genitive case: «организации» instead of «организация»

In case of English, it will be sufficient to provide just the lemmas for both the hyponym and the hypernym. In Russian, this will make the question sentence uncoordinated because the hypernym word should be present in the genitive case. Also, such words as «молоко» (milk) and «дом» (house) are written identically both in nominative and accusative cases, which causes inflection problems. This limitation has been dealt with the pymorphy2 morphological analyzer and generator [14] by estimating the most suitable word form that needs to be inflected into the genitive case according to the heuristic

[2] https://toloka.yandex.ru/

$$\mathrm{score}(w|t,c) = \mathrm{p}(t|w) + 1(t = \mathrm{noun}) \times 10 + 1(c = \mathrm{nominative}) \times 2$$

where 1(·) is the indicator function, p($t|w$) is the probability of the tag t assigned to the word w estimated on OpenCorpora [3], and c is the grammatical case. This heuristic prefers the nouns in the nominative case because the input words in the present study are in fact noun lemmas.

5.2. Dataset

The answers have been aggregated using the Yandex.Toloka proprietary answer aggregation mechanism. As the result, 4,576 out of 10,600 pairs have been annotated as positive while the rest 6,024 have been annotated as negative. Interestingly, in average, the workers were more confident in negative answers rather than in the positive ones according to the two-tailed t-test with the significance level of 0.01. These negative answers are extremely useful for both training and testing different relation extraction methods [28]. To the best of our knowledge, this is the first dataset of this kind made for the Russian language using microtask-based crowdsourcing.

5.3. Experimental Setup

Since for each of the top 300 nouns each method should provide no less and no more than five hypernyms, including the missing ones, the performance of the methods is quantitated using the precision, recall, and F_1-score. A hypernym is considered as predicted correctly if and only if it is not empty and is annotated as positive by the crowd workers.

5.4. Results

The evaluation results on LRWC showed that this resource does correlate with RuThes in terms of correctness. However, it is not necessarily a reliable source of subsumptions given the fact that it represents the human judgements, not the expert knowledge.

Table 2. Evaluation results on the LRWC dataset, top three results are highlighted

Method	Precision	Recall	F_1-score
RuThes	**0.7035**	**0.9168**	**0.7961**
Joint + Exp	0.6719	0.9002	**0.7695**
Joint	0.6726	0.8975	**0.7690**
Wiktionary + SWN	0.6287	0.8775	0.7326
Wiktionary + Exp + SWN	0.6254	0.8779	0.7304
Joint + SWN	0.5590	**0.9306**	0.6985
Joint + Exp + SWN	0.5569	**0.9304**	0.6968
RWN (Nouns)	0.5878	0.8400	0.6917
SAD + Exp	0.6313	0.6141	0.6226
SAD	0.6321	0.6121	0.6220

Method	**Precision**	**Recall**	**F_1-score**
Patterns	0.4821	0.8710	0.6207
Wiktionary + Exp	**0.7488**	0.3485	0.4756
Wiktionary	**<u>0.7492</u>**	0.3467	0.4741
Patterns + Limit	0.6711	0.3103	0.4244
Patterns + Limit + Exp	0.6700	0.3105	0.4244

RuThes showed the best results on the LRWC dataset according to F1-score and the third best result according to precision and recall. Surprisingly, the highest value of precision is yielded by the Wiktionary dataset that has been created using crowdsourcing by a group of volunteers. Although it shows relatively small recall, this observation indicates a tremendous potential of collaborative lexicography.

With expansion or without it, SWN showed the designed trade-off between precision and recall in the favor of recall, which agrees with the previous experiment (Table 1). It leaves one with a choice: it is possible to either achieve the highest recall by ignoring the information encoded by the synonyms and their common hypernyms, or to exploit this information by slightly reducing the recall while maintaining the third best value of F_1-score. Notably, on LRWC, the Joint dataset yielded better results without SWN. This is caused by the hypernymy propagation property of the method mentioned in Section 3.1, i.e., when the most frequent hypernyms overweight the others in a hierarchical context.

6. Conclusion

In this paper, Watlink, a robust method for constructing a semantic word network has been proposed. The method showed good results by outperforming competing methods on recall and F_1-score on two different datasets: an expert-built thesaurus, RuWordNet, and a new dataset representing the human judgments for Russian subsumptions, LRWC.

The implementation of the present method is available on GitHub under the MIT license: https://github.com/dustalov/watlink. The LRWC dataset is available on Zenodo in the tab-separated values format under a Creative Commons Attribution-ShareAlike license: https://doi.org/10.5281/zenodo.546302.

Acknowledgements. The reported study is funded by RFBR according to grant no. 16-37-00354 мол_а. The author is grateful to Natalia Loukachevitch who provided machine-readable versions of the RuThes and RuWordNet datasets. The author would also like to thank Alexander Panchenko, Nikolay Arefyev, Andrey Sozykin, and three anonymous reviewers for useful comments on the present study.

References

1. *Abramov N.* (1999), The dictionary of Russian synonyms and semantically related expressions [Slovar’ russkikh sinonimov i skhodnykh po smyslu vyrazhenii], Russkie Slovari, Moscow, Russia.

2. *Biemann C.* (2006), Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems, Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, New York, NY, USA, pp. 73–80.
3. *Bocharov V. V., Alexeeva S. V., Granovsky D. V., Protopopova E. V., Stepanova M. E., Surikov A. V.* (2013), Crowdsourcing morphological annotation, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", Bekasovo, Russia, pp. 109–124.
4. *Braslavski P., Ustalov D., Mukhin M., Kiselev Y.* (2016), YARN: Spinning-in-Progress, Proceedings of the 8th Global WordNet Conference, Bucharest, Romania, pp. 56–65.
5. *Dikonov V. G.* (2013), Development of lexical basis for the Universal Dictionary of UNL Concepts, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", Bekasovo, Russia, pp. 212–221.
6. *Evgen'eva A.P.* (1999), Small Academic Dictionary [Malyi akademicheskii slovar'], Rus. yaz; Poligrafresursy, Moscow, Russia.
7. *Faralli S., Panchenko A., Biemann C., Ponzetto S. P.* (2016), Linked Disambiguated Distributional Semantic Networks, The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II, pp. 56–64.
8. *Fellbaum C.* (1998), WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA, USA.
9. *Fu R., Guo J., Qin B., Che W., Wang H., Liu T.* (2014), Learning Semantic Hierarchies via Word Embeddings, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MA, USA, pp. 1199–1209.
10. *Hearst M. A.* (1992), Automatic Acquisition of Hyponyms from Large Text Corpora, Proceedings of the 14th Conference on Computational Linguistics — Volume 2, Nantes, France, pp. 539–545.
11. *Kan K.-L. and Hsueh H.-Y.* (2013), Conceptual Information Retrieval System Based on Automatically Constructed Semantic Word Network, Intelligent Technologies and Engineering Systems, Proceedings of the 2nd International Conference on Intelligent Technologies and Engineering Systems (ICITES2013), Kaohsiung, Taiwan, pp. 277–283.
12. *Kiselev Y., Porshnev S. V., Mukhin M.* (2015), Current Status of Russian Electronic Thesauri: Quality, Completeness and Availability [Sovremennoe sostoyanie elektronnykh tezaurusov russkogo yazyka: kachestvo, polnota i dostupnost'], Software Engineering [Programmnaya inzheneriya], Vol. 6, pp. 34–40.
13. *Kiselev Y., Porshnev S. V., Mukhin M.* (2015), Method of Extracting Hyponym-Hypernym Relationships for Nouns from Definitions of Explanatory Dictionaries [Metod izvlecheniya rodovidovykh otnoshenii mezhdu sushchestvitel'nymi iz opredelenii tolkovykh slovarei], Software Engineering [Programmnaya inzheneriya], Vol. 10, pp. 38–48.

14. *Korobov M.* (2015), Morphological Analyzer and Generator for Russian and Ukrainian Languages, Analysis of Images, Social Networks and Texts: 4th International Conference, AIST 2015, Revised Selected Papers, Yekaterinburg, Russia, pp. 320–332.
15. *Lee S., Lee M., Kim P., Jung H., Sung W.-K.* (2010), OntoFrame S3: Semantic Web-Based Academic Research Information Portal Service Empowered by STARWIN, The Semantic Web: Research and Applications: 7th Extended Semantic Web Conference, ESWC 2010, May 30—June 3, 2010, Proceedings, Part II, Heraklion, Crete, Greece, pp. 401–405.
16. *Loukachevitch N. V.* (2011), Thesauri in Information Retrieval Tasks [Tezaurusy v zadachakh informatsionnogo poiska], Idz-vo MGU, Moscow, Russia.
17. *Loukachevitch N. V., Lashevich G., Gerasimova A. A., Ivanov V. V., Dobrov B. V.* (2016), Creating Russian WordNet by Conversion, Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue", Moscow, Russia, pp. 405–415.
18. *Lyashevskaya O., Sharoff S.* (2009), Frequency dictionary of modern Russian based on the Russian National Corpus [Chastotnyi slovar' sovremennogo russkogo yazyka (na materialakh Natsional'nogo korpusa russkogo yazyka)], Azbukovnik, Moscow, Russia.
19. *Manning C. D., Raghavan P., Schütze P.* (2008), Introduction to Information Retrieval, Cambridge University Press, Cambridge, UK.
20. *Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J.* (2013), Distributed Representations of Words and Phrases and their Compositionality, Advances in Neural Information Processing Systems 26, Harrahs and Harveys, NV, USA, pp. 3111–3119.
21. *Navigli R., Ponzetto S. P.* (2012), BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artificial Intelligence, Vol. 193, pp. 217–250.
22. *Panchenko A., Morozova O., Naets, H.* (2012), A Semantic Similarity Measure Based on Lexico-Syntactic Patterns, Proceedings of KONVENS 2012, Vienna, Austria, pp. 174–178.
23. *Panchenko A., Ustalov D., Arefyev N., Paperno D., Konstantinova N., Loukachevitch N., Biemann C.* (2017), Human and Machine Judgements for Russian Semantic Relatedness, Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016, Revised Selected Papers, Yekaterinburg, Russia, pp. 303–317.
24. *Riedl M., Biemann C.* (2016), Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, pp. 617–622.
25. *Sanchez-Monzon J., Putzke J., Fischbach K.* (2011), Automatic Generation of Product Association Networks Using Latent Dirichlet Allocation, Procedia — Social and Behavioral Sciences, Vol. 26, pp. 63–75.
26. *Shwartz V., Goldberg Y., Dagan, I.* (2016), Improving Hypernymy Detection with an Integrated Path-based and Distributional Method, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, pp. 2389–2398.

27. *Sowa J. F.* (1987), Semantic Networks, available at: http://www.jfsowa.com/pubs/semnet.htm
28. *Ustalov D., Arefyev N., Biemann C., Panchenko A.* (2017), Negative Sampling Improves Hypernymy Extraction Based on Projection Learning, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, pp. 543–550.
29. *Ustalov D.* (2017), Concept Discovery from Synonymy Graphs [Obnaruzhenie ponyatii v grafe sinonimov], Computational Technologies [Vychislitel'nye tekhnologii], Vol. 22, Special Issue 1, pp. 99–112.
30. *van Dongen S.* (2000), Graph Clustering by Flow Simulation, Ph.D. thesis, University of Utrecht, Utrecht, The Netherlands.
31. *Zesch T., Müller C., Gurevych I.* (2008), Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary, Proceedings of the 6th International Conference on Language Resources and Evaluation, Marrakech, Morocco, pp. 1646–1652.

MULTI-LEVEL STUDENT ESSAY FEEDBACK IN A LEARNER CORPUS[1]

Vinogradova O. I. (olgavinogr@gmail.com)[1],
Lyashevskaya O. N. (olesar@yandex.ru)[1,2],
Panteleeva I. M. (irapanteleeva@rambler.ru)[1]

[1]National Research University Higher School of Economics,
[2]Vinogradov Institute of the Russian Language RAS, Moscow, Russia

The paper presents the results of using computer tools and of designing an inspection program for the purposes of the automated and semi-automated syntactic, lexical, and grammar error analysis of student essays in a learner corpus. The texts in the corpus were written in English by Russian learners of English. In our experiment we compare the parameters of the essays graded by professional examiners as the best and those graded the lowest in the pool of about 2000 essays. At the first stage in the experiment we applied a syntactic tool for parsing the sentences and collected data regarding mean sentence depth and the average number of relative, other adnominal, and adverbial clauses, then analyzed the results of lexical observations in those texts (such as average word length, number of academic words, number of linking words and some others), and finally collected the statistics related to the errors pointed out in manual expert annotation. The parameters that had very different values for the “good” and for the “bad” essays are regarded by the authors as worthy parts of the feedback a student can get for the text uploaded into the learner corpus.

Key words: learner corpora, corpus research, essay evaluation, automated feedback, lexical complexity, syntactic complexity

АВТОМАТИЗИРОВАННЫЙ КОММЕНТАРИЙ К ЭССЕ В УЧЕБНОМ КОРПУСЕ

Виноградова О. И. (olgavinogr@gmail.com)[1],
Ляшевская О. Н. (olesar@yandex.ru)[1,2],
Пантелеева И. М. (irapanteleeva@rambler.ru)[1]

[1]Национальный Исследовательский университет «Высшая школа экономики», [2]Институт русского языка им. В. В. Виноградова РАН, Москва, Россия

[1] The article was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2016–17 (grant № 16-05-0057 “Learner corpus REALEC: Lexicological observations”) and by the Russian Academic Excellence Project “5-100”.

В статье представлены результаты применения компьютерных инструментов оценки лексического и грамматического уровня текстов для автоматического или полуавтоматического анализа студенческих эссе в обучающем корпусе. Тексты в корпусе написаны на английском языке русскими студентами, изучающими английский язык. В нашем исследовании мы сравниваем разные параметры примерно двух тысяч эссе, которые оценены экзаменаторами как лучшие и как худшие работы. На первом этапе мы применили синтаксический инструмент для разбора предложений и собирали данные о средней глубине предложения и среднем количестве разных типов придаточных предложений, затем проанализировали результаты работы лексического инспектора (например, данные о средней длине слова, количестве слов из научного лексикона, количестве связующих слов), и, наконец, собрали статистику, связанную с ошибками, указанными в ручной экспертной аннотации. Параметры, существенно разнящиеся в «хороших» и «плохих» эссе, предполагается включить в форму, которую студент будет получать в режиме обратной связи после загрузки своей работы в корпус.

Ключевые слова: учебные корпуса, корпусные исследования, оценка эссе, автоматическая оценка текста, лексическая сложность, синтаксическая сложность

1. Introduction

It has many times been demonstrated over more than 20 years of learner corpora research that access to a learner corpus contributes greatly to the efficiency of L2 acquisition for both learners and instructors alike (Granger 2012; Granger et al. 2013). REALEC is a learner resource which has been in active use by English instructors teaching at the university level. It is the first collection in the open access of English texts written by Russian university students learning English. It is available with all errors in the student essays outlined by expert annotators (Vinogradova 2016). This paper looks at the syntactic complexity and at the lexical diversity range in the best essays of the past examination in comparison with those that were considered the worst among the examination essays written by the university students in a 2015 administration of the 2nd-year examination in English. This paper aims at evaluating which features constitute the indications of successful / unsuccessful text and can thus be included in automatic essay feedback that a student can get after uploading his / her essay in the corpus.

2. Related work

Published in 2015 Cambridge Handbook of Learner Corpus Research includes a few papers describing approaches to providing learners of a second/foreign language with automatic commentary on the quality of their written production. The papers with the focus on or related to the lexical features of the student text are Adel, 2015 and Granger, 2015. Tom Cobb and Marlise Horst in their chapter of Cambridge Handbook of Learner Corpus Research spoke about the generalizing role of a learner corpus in shedding light on second language acquisition by allowing the use of many computing tools inapplicable to separate texts (Cobb & Horst, 2015, pp. 185–206)

The choice of lexical parameters to be included in evaluation is discussed, for example, in Lavallée & McDonough, 2015. The adjacent filed—comparisons of student texts with authentic academic texts—were reported by researchers from University of Grenoble-II in their work which presents Apex, a system for automatic assessment of a student essay based on the use of Latent Semantic Analysis (Dessus & Lemaire, 2001). McCarthy & Jarvis, 2010 report the comparative assessment of different lexical features in the process of automated evaluation.

Application of syntactic parsing in corpus studies is the topic of many a work of the recent years. Many publications of authors working in cooperation with Daniella McNamara (like McNamara et al. 2011) relate diagnostics of advanced measures of linguistic complexity of a text to the application of an automated tool called Coh-Metrix designed to assess the characteristics of texts for different purposes, and syntactic sophistication is just one of them. The syntactic complexity analyzer by (Lu, Ai 2015), which provides a set of simple yet detailed measurements such as the mean length of clause, the number of dependent clauses and coordinate phrases, has currently become a state-of-the-art benchmark, although the need for more sophisticated measures is discussed in the professional community. The report of the TREACLE project with its reference to the use of the Stanford Parser in a learner corpus of works written by Spanish learners of English (Murcia-Bielsa & MacDonald 2013, page 337) became one of the starting points for our experiment.

The vast literature on the parameters of student writing used in pedagogical expert evaluation has been discussed for examinations of different types administered by different institutions. In view of situation with the English examination at the Higher School of Economics, we have chosen for reference one of the recent and most detailed reports that consider writing potential indications in IELTS examination (Cotton & Wilson 2011). According to this report, the four parts of the grade assigned by examiners are measured by looking at:

- the number of words, relevance to the topic in the question, and coverage of all parts of the question (Task Achievement/Task Response);
- organisation in paragraphs, connection of sentences and paragraphs with logical links and referencial tools, no or little repetition (Coherence and Cohesion);
- use of appropriate academic words and collocations, use of paraphrase to avoid repetition, correct spelling (Lexical Resource);
- use of a variety of grammatical forms, combination of short and complex sentences, and not too many grammatical mistakes (Grammatical Range and Accuracy).

The parameters outlined in this work have defined our selection of the features to be included in the experiment.

3. Experiment setup

The objective of our corpus experiment was to establish the correlation between the grades that examination essays were given by experts, on the one hand, and the indices of the automated analyses of student texts from a learner corpus, on the other, with the more distant goal of outlining the best features for automated essay feedback. The experiment was carried out over essays in IELTS format[2] written by 2nd-year Bachelor students in their final English examination in 2015. The writing part of this examination includes two tasks requiring that each testee writes one essay not less than 150 words long (essay1), the other about 250 words long (essay2), both within the period of one hour. The essays are assessed just as was stated in Section 2—by the following criteria: task response, coherence and cohesion, lexical resource, grammatical range and accuracy. The tasks were given to almost a thousand students. After the examination, the essays were evaluated by independent EFL raters, who assigned each task a holistic grade in the percentage points up to 100. When the essays were uploaded to REALEC, expert annotators spotted errors in the essays and added manual annotations clasifying those errors. For the purposes of the experiment, two groups of essays were chosen out of almost two thousand essays—those that the experts graded at 75% and over (33 essays), and those that got the grade of 30% and lower (43 essays). Essays in either group were subjected to the three stages of analytical procedures: 1) POS and dependency parsing; 2) automatic evaluation of each text using the built-in lexical tool REALEC-Inspector designed at the School of Linguistics, Higher School of Economics; and 3) statistical analysis of the expert annotation. The results of all three stages in two groups were compared with each other, and the conclusions are reflected upon in the final section of the paper.

4. Data analysis

4.1. POS and syntactic parsing

The sentences were processed with the open-source tagger and dependency parser UDpipe (Straka 2015). Each word was tokenized and tagged for POS and dependency types, so that the depth of the tree was easy to calculate for the sentence. For each essay, the average syntactic depth was counted with the maximum and minimum depths stated. The efficiency of UDpipe on REALEC essays was comparable to that achieved with the Stanford parser in (Murcia-Bielsa, MacDonald, 2013): the cases of wrongly defined arcs (unlabeled attachment) were minimal and could mainly be accounted for by a learner-driven gaps in syntactic structures and by distant dependencies. As for dependency relations, we only took into account the following labels: relative clauses (acl:relcl), groups with participles as the head (acl), and adverbial clauses (advcl). These relations were checked manually for false positives.

The mean syntactic depth of the sentence ranges from 1 (no more than one dependency down from the sentence head, as in *It is wrong!*) to 10. The analysis has revealed insignificant difference between the best and worst essays in their average depth (best: mean = 4.61, sd = 1.66; worst: mean = 4.12, sd = 1.57). The amount of particular subordinate clause types per essay, on the contrary, differs significantly between the best and worst essays, see Table 1 (mean and 95% CI values are shown).

[2] IELTS (International English Language Testing System) is a test of English language profi-ciency for non-native speakers of English. IELTS certificates are recognized in more than 120 countries round the world and cover all four language skills—listening, reading, writing, and speaking.

Table 1 Subordinate clause types per essay

Grade Cat	mean.acl	mean.acl:relcl	mean:advcl
best	3 ± 0.82	3.25 ± 0.85	5.41 ± 1.07
worst	1.21 ± 0.42	1.43 ± 0.38	1.86 ± 0.5

Table 2 shows Pearson's pairwise correlation between the following factors: expert's grade (absolute values), mean sentence depth, number of adnominal clauses including participle groups (N_acl), number of relative clauses (N_acl:relcl), number of adverbial clauses (N_advcl), total number of subordinate clauses. The correlation between the number of relative and adverbial clauses is moderate, while the acl score behaves differently. Furthermore, it is predictable that neither of these features correlate with the mean syntactic depth. The amount of adverbial clauses correlates best with the grade since they are used more frequently, and it shows the students' ability to express a variety of causal, temporal, and other relations between propositions.

Table 2 Correlation of the syntactic features

	Mean Depth	N_acl	N_acl: relcl	N_advcl	N_All SubordCl
Grade	0.203	0.397	0.462	0.599***	0.630
MeanDepth		0.375	0.311	0.179	0.346
N_acl (adnominal clauses)			0.355	0.383	0.698
N_acl:relcl (relative clauses)				0.548	0.785
N_advcl (adverbial clauses)					0.867

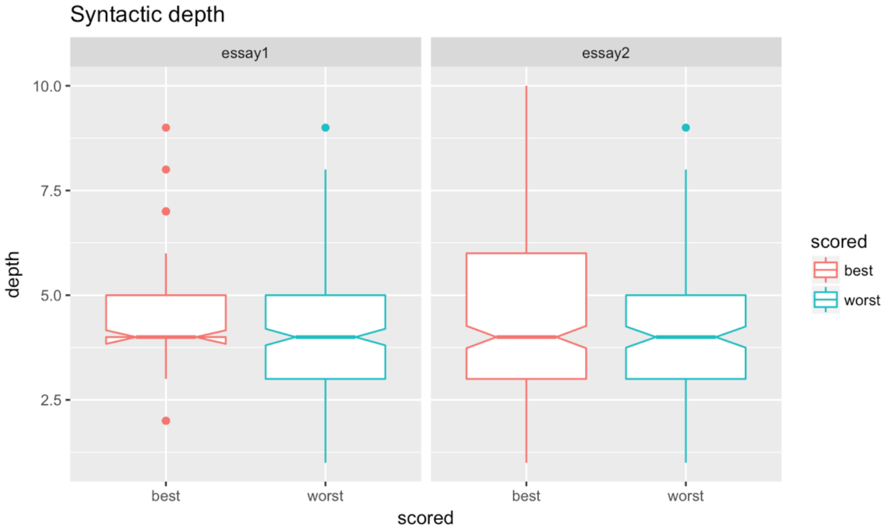


Fig. 1. Mean syntactic depth of sentences by essay type and grade category

Thus, these features should constitute the basis of the student automatic feedback, although more detailed analysis is needed.

4.2. Lexical evaluation with REALEC-Inspector

While considering which parameters to take up for the lexical inspection, we decided to start with those that have been described by the authors working in corpus linguistics as automatically indicative of the level of lexical variety. McCarthy & Jarvis, 2010 pointed out the importance of the length of words and length of sentences as the criteria for the automated lexical evaluation. Frequency of a word in the Corpus of Contemporary American English [3] was justified as a parameter in lexical evaluation in Crossley, Cobb, & McNamara, 2013 and Vongpumivitch, Huang, & Chang, 2009, while for checking the use of academic vocabulary the three lists have been argued in corpus linguistics works: the Coxhead Academic Word List (cf. Coxhead, 2000 and Coxhead, 2011), the one in the Corpus of Contemporary American English[4] and the Pearson academic collocation list[5]. That was why all those parameters were included in our experiment:

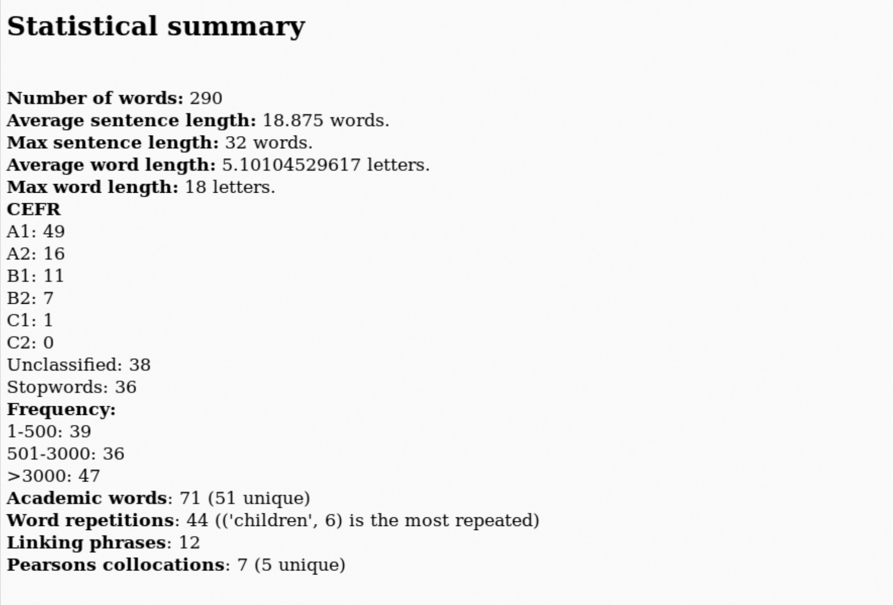
1) Number of words in the essay
2) Average length of a sentence in the essay
3) Length of the longest sentence in the essay
4) Average length of word in the essay
5) Length of the longest word in the essay
6) Number of words of each level of CEFR in the essay
7) Number of words from the COCA frequency lists
8) Number of academic words in the essay with repetitions and without them
9) Number of repetitions of words used in the essay. The word most frequently repeated.
10) Number of linking words and expressions in the essay

For the purposes of the experiment and for the broader perspectives of providing automated lexical analysis for any learner text, we developed the application REALEC-Inspector. Its homepage was placed in the Moodle environment with the options either to upload a text in an input window, or browse for the text in REALEC. For the indices listed above the inspection of the text with REALEC-Inspector opens right after the text with the short statistical summary, of which a sample is given in Figure 2.

[3] COCA (The Corpus of Contemporary American): http://corpus.byu.edu/coca/, http://www.wordfrequency.info/, http://www.academicvocabulary.info/

[4] http://www.academicvocabulary.info/

[5] http://pearsonpte.com/research/academic-collocation-list/



Statistical summary

Number of words: 290
Average sentence length: 18.875 words.
Max sentence length: 32 words.
Average word length: 5.10104529617 letters.
Max word length: 18 letters.
CEFR
A1: 49
A2: 16
B1: 11
B2: 7
C1: 1
C2: 0
Unclassified: 38
Stopwords: 36
Frequency:
1-500: 39
501-3000: 36
>3000: 47
Academic words: 71 (51 unique)
Word repetitions: 44 (('children', 6) is the most repeated)
Linking phrases: 12
Pearsons collocations: 7 (5 unique)

Fig. 2. List of statistics for the essay under inspection

The statistical summary is then followed by detailed comments for each item on the list, and for some of them the Inspector provides diagrams. Here we show some of them giving the sources of the reference materials applied.

For the histogram of CEFR words distribution (Fig. 3), Word Family Framework was used (the possibility to use English Vocabulary Profile instead has been reserved), and each word—with the exception of stop words (there are 153 of them in the application)—is lemmatized with the help of NLTK. Words that the system was unable to relate to a particular CEFR level (among them are misspelled words) are categorized as "Unclassified" and given on the histogram in column 0.

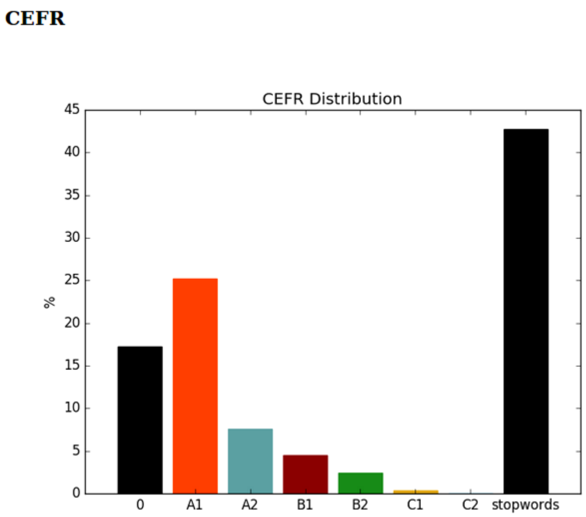


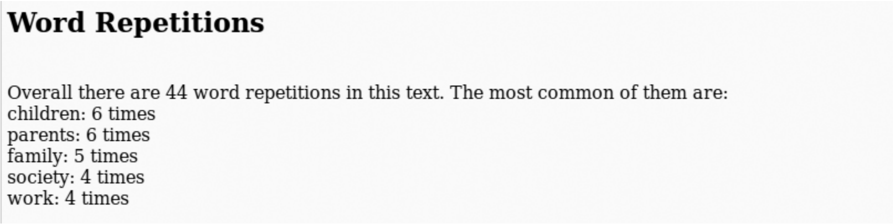
Fig. 3. Sample picture of distribution of CERF-level words

After that the author of the text gets the list of words from the essay that are among the 500 most frequent words in COCA, and then those that are among the 3,000 most frequent words in COCA. Stop-words are again excluded.

The next comment is on the occurrence of academic words from the list which is a combination of two—the Academic Word List Coxhead and the Corpus of Contemporary American English List of Academic Words. As a result, if a word from an essay belongs to either of these lists, it will be counted.

In this section the author will see a diagram showing the distribution of the number of academic vocabulary items across all essays in the corpus, with the red line marking the average index in the essay under inspection for the author to compare with other essays. Useful as it may be, we don't bring in an example of the diagram here, as it goes beyond the scope of this experiment to research whether the comparison with all essays in the corpus gives students the way to understand where their writing stands as far as sophistication is concerned.

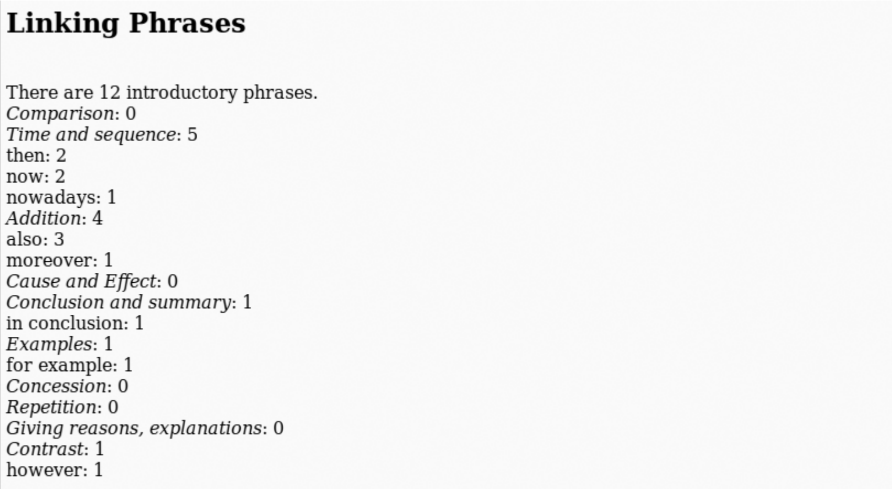
Next, five most frequently repeated words (those that are not stop-words) are shown (see Figure 4). The need for demonstrating the ability to paraphrase can be emphasized here.


Word Repetitions

Overall there are 44 word repetitions in this text. The most common of them are:
children: 6 times
parents: 6 times
family: 5 times
society: 4 times
work: 4 times

Fig. 4. Sample list of repetitions in the essay

The number and the list of linking words and introductory expressions used in the essay are accompanied at the next stage next by the indication of their categories (Comparison, Time and sequence, Addition, Cause and Effect, Conclusion and summary, Examples, Concession, Repetition, Giving reasons, Explanations, Contrast (Figure 5).



Linking Phrases

There are 12 introductory phrases.
Comparison: 0
Time and sequence: 5
then: 2
now: 2
nowadays: 1
Addition: 4
also: 3
moreover: 1
Cause and Effect: 0
Conclusion and summary: 1
in conclusion: 1
Examples: 1
for example: 1
Concession: 0
Repetition: 0
Giving reasons, explanations: 0
Contrast: 1
however: 1

Fig. 5. Sample list of linking words in the essay

The comparison of the use of linking phrases in the essays under inspection with all other essays in the corpus can also be presented to the author on the histogram as an additional option, see Figure 6.

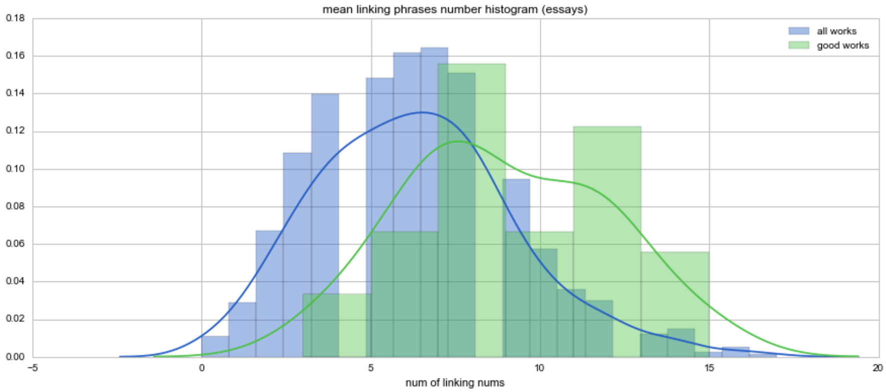
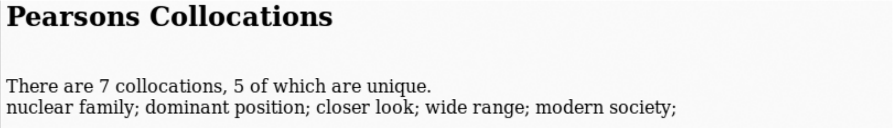


Fig. 6. Distribution of linking phrases number: good essays vs all essays

The inspector then gives the number and the list of collocations from the essay if they are on the Pearson Academic Collocation List (see Figure 7).



Pearsons Collocations

There are 7 collocations, 5 of which are unique.
nuclear family; dominant position; closer look; wide range; modern society;

Fig. 7. Sample list of collocation in the essay

There is also an option to ask for the visualization of the text with one of the three features:

1) with words of different CEFR levels presented in different colours;
2) with words of different COCA frequencies presented in different colours;
3) with academic words highlighted.

Table 4 below gives the summary of the comparative analysis of lexical features under investigation for the two sets of essays (the best and the worst).

Table 4. Synopsis of the comparison between the experimental sets

Parameters for automated lexical inspection	**Essays scored 75% and higher**		**Essays scored lower than 30%**	
	Task 1	**Task 2**	**Task 1**	**Task 2**
1) Number of words in the essay	203	292	174+	161***
2) Average length of a sentence in the essay	21	20	17^{ns}	16^{ns}
3) Length of the longest sentence in the essay	37	39	33^{ns}	30^{ns}
4) Number of academic words in the essay (with repetitions/without repetitions)	41/28	69/51	33^{ns}/18+	42***/ 29***
5) Number of linking words and expressions in the essay	5	7	3^{ns}	4^{ns}
6) Number of collocations from the Pearson academic collocation list in the essay (with repetitions/ without repetitions)	0,8/0,8	0,73/0,73	0,38***/ 0,35***	0,38***/ 0,38***

The parameters that have not been included in the table are those whose values were approximately the same for essays scored highly and for those scored very low: the average word length, maximum word length, number of word repetitions, words of different CEFR levels. In general, "good" essays have more CEFR scale words at each level, as well as more words of high frequency in COCA, but not many more of them. This is rather due to the fact that texts showing better writing proficiency have higher overall number of words. So, the figures have not been included in the comparison.

It is clear from the table that the best characteristics distinguishing texts that are more likely to get a good score from those that are less so are the following:

- average sentence length;
- the number of words from academic vocabulary lists;
- the number of academic collocations.

As these parameters can be evaluated by a software application, they will be included in the automated feedback provided to authors of learner texts.

4.3. Error analysis

Error annotation in REALEC is based on the classification scheme of about 150 specific error tags organized into a tree-like structure with 7 classes of errors—Spelling, Capitalisation, Grammar (Morphology), Grammar (Syntax), Vocabulary and Discourse. The overall number of errors spotted by the annotators varies both in the "best" and "worst" essays, and it can be explained by the following consideration: authors with stronger writing potential make more effort to apply sophisticated morphological and syntactic features than those with less proficiency, and as a result the former run a greater risk of making mistakes than the latter. The approach in the examination of IELTS type is to encourage attempt at higher sophistication rather than penalize incorrectness in complicated constructions, either grammar or vocabulary, so the first group of authors get higher grades more often than the second. On the other hand, weaker writers are more prone to making mistakes in simple cases than those with better writing skills. And these two opposing arguments lead to the situation, in which the average numbers of errors in an essay is not a good indicator of the writing proficiency, nor does the average number of syntactic and/or discourse errors demonstrate the level of syntactic complexity of the text. The tagging statistics across the "best" and the "worst" essays within the scope of our research shows exactly the same distribution in Table 5.

Table 5. Error annotation indices in the experimental folders

	Essays scored 75% and higher	**Essays scored lower than 30%**
Average number of all error tags in one essay	19	19.5
Minimum and the maximum number of all error tags	3 to 60	10 to 66
Average number of syntactic tags	2	3
Average number of discourse tags	3	3

The two possible ways of demonstrating annotation results in the feedback are either to give the overall number of tags in comparison with the average number across the folder with similar essays, as well as the number of the tags for repeated categories of errors in the essay, also against the average in the folder, or just summarise the numbers like this:

> The expert pointed out 14 errors (19 average) (5 syntactic, 4 discourse errors, and 3 morphological). You may need to review the use of different syntactic constructions.

automated tagging as well.

5. Conclusions

The observations over the features of many student essays in the learner corpus have confirmed the following points important for working out approaches to automated evaluation of student writing and to automated feedback for student writing:

- word length and number of repetitions are insignificant as indicators of the writing proficiency.;
- the numbers of words at each CEFR level and of those with high COCA frequency are to some extent larger in essays highly evaluated by experts, but their relevance as a part of automated feedback has to be confirmed further. Dependence of the lexical variety and complexity on the length of a piece of writing has many times been emphasized in corpus linguistic research, but the texts of essays at our disposal were of two types—not less than 150 and not less than 250 words, so for the purposes of our experiment all statistical analysis in the lexical inspection was carried out separately for the two types of essays—descriptions of the illustration(s) given in the task and argumentative essays.

The results of the comparison allow us to state that automatic application of both syntactic parsing and lexical inspection will provide good suggestions for improving students' writing potential and can be considered to be good predictions of the success/failure in the examination.

To cater for those users who may look for independent training, we are thinking of giving the results of the reported research one more use in a way of introducing a few computational modules that will show a user the basic characteristics of the text he/she is composing right in the process of typing in an essay, namely, instant demonstration of such features as superfluous repetitions, misspelled words, low variability of syntactic constructions, and some others.

References

1. *Annelie Ä.* (2015), Variability in learner corpora, In Granger S., Gilquin G., Meunier F. (eds.), The Cambridge Handbook of Learner Corpus Research, Cambridge, UK: Cambridge University Press, pp. 401–422.
2. *CEFR* (2001), The Common European Framework of Reference for Languages: Learning, Teaching, available at: https://www.coe.int/t/dg4/linguistic/Source/Framework_EN.pdf
3. *Cobb T., Horst M.* (2015), Learner corpora and lexis, In Granger S., Gilquin G., Meunier F. (eds.), The Cambridge Handbook of Learner Corpus Research, Cambridge, UK: Cambridge University Press, pp. 185–206.
4. *Cotton F., Wilson K.* (2011), An investigation of examiner rating of coherence and cohesion in the IELTS Academic Writing Task 2, IELTS research reports. Vol. 12, IELTS Australia and British Council, pp. 1–76.
5. *Coxhead A.* (2000), A new academic word list, TESOL Quarterly, Vol. 34 (2), pp. 213–238
6. *Coxhead A.* (2011), The academic word list 10 years on: Research and teaching implications, TESOL Quarterly, Vol 45 (2), pp. 355–362.

7. *Crossley S. A., Cobb T., McNamara D. S.* (2013), Comparing count-based and band-based indices of word frequency: Implications for active vocabulary research and pedagogical applications, System, Vol. 41 (4), pp. 965–981.
8. *González-López S., López-López A.* (2015), Lexical analysis of student research drafts in computing, Computer Applications in Engineering Education, Vol. 23 (4), pp. 638–644.
9. *Granger S.* (2012) How to use Foreign and Second Language Learner Corpora, Research Methods in Second Language Acquisition: A Practical Guide, Blackwell, Oxford. Ch.2, pp. 5–29.
10. *Granger S.* (2015), The contribution of learner corpora to reference and instructional materials design, The Cambridge Handbook of Learner Corpus Research, Cambridge, UK: Cambridge University Press, pp. 485–510
11. *Granger S., Gilquin G., Meunier F.* (eds.) (2013), Twenty Years of Learner Corpus Research. Looking Back, Moving Ahead, Proceedings of the First Learner Corpus Research Conference, Vol. 1, Presses universitaires de Louvain.
12. *Lavallée M., McDonough K.* (2015), Comparing the Lexical Features of EAP Students' Essays by Prompt and Rating, TESL Canada Journal, Vol 32 (2), pp. 30–44.
13. *Lemaire B., Dessus P.* (2001), A System to Assess the Semantic Content of Student Essays, Journal of Educational Computing Research, Vol. 24(3), pp. 305–320.
14. *Lu X., Ai H.* (2015), Syntactic complexity in college-level English writing: Differences among writers with diverse L1 backgrounds, Journal of Second Language Writing, Vol. 29, pp. 16–27.
15. *McCarthy P. M., Jarvis S.* (2010), MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment, Behavior Research Methods, 42, no 2, pp. 381–392.
16. *McNamara D. S., Graesser A. C., Cai Zh., Kulikowich J. M.* (2011), Coh-Metrix Easability Components: Aligning Text Difficulty with Theories of Text Comprehension, Proceedings of the Annual meeting of the American Educational Research Association (AERA 2011), New Orleans, LA, available at: https://www.researchgate.net/publication/228455723_Coh-Metrix_Easability_Components_Aligning_Text_Difficulty_with_Theories_of_Text_Comprehension
17. *Murcia-Bielsa S., MacDonald P.* (2013), The TREACLE project: Profiling learner proficiency using error and syntactic analysis, Twenty Years of Learner Corpus Research. Looking Back, Moving Ahead, Proceedings of the First Learner Corpus Research Conference, Presses universitaires de Louvain.
18. *Straka M., Hajič J., Straková J.* (2015), UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing, Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16), 23–28 May 2016, Portorož, Slovenia, pp. 4290–4297, UDpipe page available at: http://lindat.mff.cuni.cz/services/udpipe/run.php
19. *Verhelst N., Van Avermaet Piet, Takala S., Figueras N., North B.* (2009), Common European Framework of Reference for Languages: learning, teaching, assessment, Cambridge University Press.

20. *Vinogradova O.* (2016), The Role and Applications of Expert Error Annotation in a Corpus of English Learner Texts, Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2016" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2016"], Moscow, Russia, pp. 740–751
21. *Vongpumivitch V., Huang J. Y., Chang Y. C.* (2009), Frequency analysis of the words in the Academic Word List (AWL) and non-AWL content words in applied linguistics research papers, English for Specific Purposes, Vol. 28 (1), pp. 33–41.

AUTOMATIC COLLOCATION EXTRACTION: ASSOCIATION MEASURES EVALUATION AND INTEGRATION

Zakharov V. P. (v.zakharov@spbu.ru)

Saint-Petersburg State University, Saint-Petersburg, Russia

The paper deals with collocation extraction from corpus data. A collocation is meant as a special type of a set phrase. Many modern authors and most of corpus linguists understand collocations as statistically determined set phrases. The above approach is the basic point of this paper which is aimed at evaluation of various statistical methods of automatic collocation extraction. There are several ways to calculate the degree of coherence of parts of a collocation. A whole number of formulae have been created to integrate different factors that determine the association between the collocation components. Usually, such formulae are called association measures.

The experiments are described which objective was to study the method of collocation extraction based on the statistical association measures. We extracted collocations for the word *вода* (water) and some others by means of the tool Collocations of the NoSketch Engine system using 7 association measures. It is important to stress that the experiments were conducted using representative corpora, with large amount of the resulting collocations being under study. The data on the measure precision allows to establish to some degree that in cases when collocation extraction is not used for some special purposes such measures as *MI.l-og_f, log-Dice*, and *minimum sensitivity* should be used. No measure is ideal, which is why various options of their integration are desirable and useful. And we propose a number of parameters that allow to rank collocates in an integrated list, namely, an average rank, a normalised rank and an optimised rank.

Key words: collocation extraction, association measures, evaluation, average rank, normalised rank, optimised medium rank

АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ КОЛЛОКАЦИЙ: СРАВНЕНИЕ И ИНТЕГРАЦИЯ МЕР АССОЦИАЦИИ

Захаров В. П. (v.zakharov@spbu.ru)

Санкт-Петербургский государственный университет, Санкт-Петербург, Россия

1. Introduction: Collocations and Collocability

According to the theory of signs, the language system is based on two main types of relations between language units: paradigmatic and syntagmatic ones. Syntagmatic relations are based on linear speech (text) nature and occur on all language levels, though they are mostly studied in the framework of lexicology and syntax. When constructing a sentence, the choice of words is determined not only by their denotative and significative meanings, but rather it depends on the surrounding words which they are grammatically and semantically related to. The combinatory ability of language units, collocability, is one of the linguistic laws. However, these laws have not been studied in depth yet.

Let's speak now about the notion of collocation. There are different approaches to this term. Sometimes a collocation is meant as a synonym of a word combination, sometimes it is a special type of a set phrase. S. Evert suggests the following definition: "A collocation is a word combination whose semantic and syntactic properties can't be fully predicted on the basis of information about its constituents and which therefore should be added to the dictionary (lexicon)" [Evert 2004: 17]. However, explanatory dictionaries do not always consistently reflect the information about collocations. Perhaps, this is not even possible for a print dictionary of the whole language. Besides, the boundary between free and set phrases is quite vague. There are many set phrases whose meaning is equal to the sum of the meanings of their constituents, despite the fact that such phrases function as a single unit, with the stability rather than idiomatic nature being the main feature. A threshold of stability should be chosen to range them, above which a word combination can be called a set phrase. This approach assumes a probabilistic nature of collocations. Many modern authors and most of corpus linguists understand collocations as statistically determined set phrases. In this case, not only phrasemes and idioms, but also multiword terms, named entities (real-world objects, such as persons, locations, organisations, products, etc.,) and other types of free combinations could be regarded as set phrases.

The above approach is the basic point of our paper which is aimed at evaluation of various statistical methods of automatic collocation extraction.

2. Statistical background: association measures

Nowadays, there are several ways to calculate the degree of coherence of parts of a collocation. It is only natural to assume that one of the ways to identify the stability of a word combination is the frequency of their co-occurrence. The co-occurrence, in its turn, is associated with the frequency of individual components of the collocation. A whole number of formulae have been created (or borrowed from other sciences) to integrate different factors that determine the association between the collocation components. Usually, such formulae are called association measures. Most of them are based on the frequency comparison for pairs of words obtained by means of an actual corpus with relative frequencies taken from a hypothetical corpus where all words from the actual one are randomly located. It is carried out to identify the statistically significant fluctuations between the observed and expected frequencies [Church, Hanks 1991, Dunning 1993, Sinclair 1996, Evert., Krenn 2001].

There are different measures based on the calculation of the degree of words nearness in a text. P. Pecina provides 82 measures, describes their mathematical foundations including their formulae and key references [Pecina, 2009: 44–45, 48] (see also dissertation of S. Evert [2004]).

One should not forget also that words which tend to collocate with each other cannot be found in a random order in any case, as there exist grammar rules which imply that "the language system is a probabilistic one and it is a grammatical probability that word frequency shows in a text" [Halliday 1991: 31]. There are also methods that take into account the syntactic nature of collocations. B. Daille claims that the linguistic knowledge drastically improves the quality of stochastic systems [Daille 1994: 192]. One of the methods to take syntax in account are the so-called word sketches, which are lists of statistical collocations, each one for each syntactic relation [Kilgarriff, Tugwell 2001; Kilgarriff, Tugwell 2002]. These syntax-based collocations are described in detail by V. Seretan [2011: 59–101].

But in this paper, the grammatical probability is not taken into consideration, only the statistical one.

3. Association measures functionality

Lexical association measures being applied to a key word (node) occurrence and context statistics extracted from the corpus for all collocation candidates result in their association scores. A list of the candidates ranked according to their association scores is the desired result of the entire process. The top of the list are word combinations that are assumed to have the greatest association with each other and, consequently, be the most probable collocation candidates.

In general, all of them take into consideration the frequency of joint occurrence of a key word (node) and its collocate, thus answering the question how random the association force is between the collocates. But proper formulae are different, and they demonstrate different association strengths for the same collocations, which is why collocation ranks obtained by different measures do not coincide. It seems interesting and useful to try to reveal the functionality of different measures. It is known, too, that some measures bring similar results and others are significantly different [Křen 2006: 246–247].

The research on and evaluation of various association measures has been done for quite a long time and has been quite intensive [Dunning 1993; Evert, Krenn 2001; Braslavskij, Sokolov 2006; Pecina, 2009]. It is known that *t-score* extracts most frequent collocations. *Log-likelihood* was eventually preferred for its good behaviour on all corpus sizes and also for promoting less frequent candidates. On the contrary, the MI measure allows to reveal low-frequency multiword terms and proper names. Furthermore, it should be noted that the raw frequency of pairs was also found to be a good indicator of termhood, but it has the disadvantage of not being able to identify rare terms [Daille, 1994: 172–173].

Besides, association score depends on the type of the units (lemmas or word forms) whose statistics are used for the calculations. Sometimes collocation extraction by statistical measures has to be done on the word form level rather than on the lemma level. The analysis described in [Zakharov, Khokhlova 2014: 340] has shown that in some cases word form collocations overwhelmingly have significantly bigger value for all association measures.

The very number of the calculated collocates and the value of the association measure are also dependent on the "window" between the node and the collocate that has been chosen for the calculations.

4. Collocation extraction: integration of different association measures

The experiments were conducted on the basis of the Araneum corpora of Russian (http://unesco.uniba.sk), with the access provided through the NoSketch Engine. These corpora belong to the family of web corpora being created by the wacky technology [Benko 2014; Benko, Zakharov 2016]. We used 2 corpora, Russicum Russicum Minus (120 mln. tokens), Russicum Russicum Maius (1,20 bln.).Both consist of texts downloaded from the .ru domain sites. The access to corpora is provided through the NoSketch Engine [Rychlý 2007].

We extracted collocations for the words *вода* (water), *враг* (enemy) and *рыба* (fish) by means of the tool *Collocations* of the NoSketch Engine system using 7 association measures: *T-score*, *MI*, *MI3*, *log likelihood*, *minimum sensitivity*, *logDice* and *MI.log_f* [Statistics Used in Sketch Engine]. These measures are popular in many other systems, too. The major part of experiments was conducted on Russicum Russicum Minus corpus. This article provides the data obtained for the collocation window (–3, +1).

The result us represented by a list of collocates (collocations) organized for each of the 7 above association measures ranged according to the association score in the form of a table (see an example for the query *вода* (water) in Table 1). The number of collocates for each query was 200.

Table 1. List of collocates for вода (water) extracted by means of MI.log_f measure (a fragment)

Collocates	Co-occurrence count	Candidate count	MI.log_f score
Сточный (sewer)	12,479	13,791	100,505
Питьевой (drinkable)	11,288	14,006	97,878
Грунтовый (ground)	8,672	11,598	94,132
Кипяченый (boiled)	3,635	4,502	86,016
Горячий (hot)	20,665	102,240	84,393
Минеральный (mineral)	9,409	45,044	78,146
Холодный (cold)	15,172	102,915	77,386
Талый (melt)	1,863	2,701	77,295
Проточный (flowing)	2,602	5,125	77,246
Дистиллированный (distilled)	1,517	1,849	77,021
Пресный (fresh)	2,124	3,883	76,077
Подсоленной (salty)	1,082	1,211	74,331
Дождевой (rain)	2,279	5,476	73,727
Литр (litre)	9,275	63,939	73,217
Околоплодных (delivery)	767	806	71,279
Теплый (warm)	13,473	136,681	70,910
Кипеть (boil)	2,637	9,274	70,789
…………………	……	……	……

A rank has been assigned to every collocate in a table for each measure according to the score of the appropriate measure.

The next part of the study is aimed at developing methods for the integrated use of different measures of association. We used 7 collocation lists obtained in the first experiment. The ranged lists of collocates extracted by the 7 above association measures were processed in the following manner. Meaningless collocations with punctuation marks were removed. Due to errors of lemmatization, some collocates were presented in several different word forms. For such cases, non-lemmatized word forms of the same word were united into a single unit, with the highest association value being chosen. "Clean" lists of collocates were obtained as a result. Then, 7 tables (with 100 collocates in each) were merged into a new one in such a way so as to the collocates that were obtained through several measures were merged into a single line of the summary table, with their rank for each measure being provided (Table 2). When a collocate was not available among the first hundred collocates for appropriate measure it was not ranked.

Table 2. Summary table of collocates for *вода* (water)

Collocates	T-score	MI	MI3	log likelihood	min. sensitivity	log-Dice	MI.log_f
Сточный (sewer)	5	25	1	2	5	4	1
Питьевой (drinkable)	7	39	2	4	7	6	2
Грунтовый (ground)	13	53	4	7	13	10	3
…	…	…	…	…	…	…	…
Отвод (drainage)	60		35	37	64	50	29
Родниковый (sping)		78	70				30
Туалетный (cologne)	73		37	45	75	57	31
Обеззараживание (decontaminated)			47	69			32
…	…	…	…	…	…	…	…

It is clear that the same collocations with the word *вода* in the ranked list of different measures have different rank. Then a question arises: what is the rank of a certain collocation in such merged list, or, in other words, what single rank should be assigned for each collocation.

The following hypotheses were made:

1) the more the number of the measures in this combined list that identified a relevant collocate, the stronger the collocability of a given collocation;
2) the less the sum of the ranks or the average rank for a relevant collocate, the stronger the collocability, and, consequently, this sum can be regarded as the coefficient of the "value" of this collocation: the lesser sum makes a given collocation more "valuable" (potentially stronger);
3) if both conditions are observed then the "value" of a given collocation is even higher, which is why we introduce a normalised rank.

As a result, the following indicators have been added to Table 3:

1) the *number of association measures* that have “calculated” a given collocate (within 100 “cleaned” lines for each measure);
2) the *average rank* of the collocate: the sum of all ranks divided by the value “the number of association measures”;
3) the *normalised rank* of the collocate (Table 3).

Table 3. Summary table of collocates for *вода* (water)

Collocates	T-score	MI	MI3	log likelihood	min. sensitivity	log-Dice	MI.log_f	Number of measures	Avera-ge rank	Nor-malised rank
Сточный (sewer)	5	25	1	2	5	4	1	7	6.14	6.14
Питьевой (drinkable)	7	39	2	4	7	6	2	7	9.57	9.57
Грунтовый (ground)	13	53	4	7	13	10	3	7	14.71	14.71
…	…	…	…	…	…	…	…			
Отвод (drainage)	60	0	35	37	64	50	29	6	45.83	51.33
Родниковый (spring)		78	70				30	3	59.33	103.23
Туалетный (cologne)	73	0	37	45	75	57	31	6	53.00	59.36
Обеззараживание (decontamination)	0	0	47	69	0	0	32	3	49.33	85.83
…	…	…	…	…	…	…	…			

The normalised rank is derived from the average rank multiplied by the coefficient that is in inverse proportion to the number of the association measures that have “calculated” a given collocate (NB: the less the rank, the more valuable the collocation is).

The coefficient for the normalised rank is calculated by the following formula:

$$log_2(1+7/n),$$

where *n* is the number of the successful measures for this collocate.

It is safe to say that the average and the normalised ranks “objectify” (integrate) the functionality of various association measures.

It should be noted that the less the rank the stronger (in theory) is the strength of the association between the collocation components. However, the rank is determined based on association measure score, which is why it is our task to correlate the ranks, i.e. the association strength, with some truth criterion.

5. Evaluation

Usually, comparison to some “gold standard“ or expert evaluation are used to evaluate the results of automated systems. When methods of collocation extraction are evaluated both options appear to be problematic. There is no „gold standard“ that would fully or significantly cover the set phrases. We could try to build it *ad hoc* for selected key words based on various dictionaries, but, due to the incomplete nature of dictionaries, the quality would be doubtful. As to expert evaluation, it is very expensive, taking into account time and human resources. The majority of automatic methods of word combination identification use large amount of data and result in large collocation lists. It would be prudent here to mention the volume of the sample evaluated. Expert method of evaluation usually covers only a small part of data due to its labour intensity. Unfortunately, the quality of automated methods is often evaluated based on the examples taken from the top units of ranked lists, and from a small number of the resulting collocates [Seretan 2011: 70].

In this paper, we have used expert evaluation, which means that each of 247 collocations of the summary list was marked either as a set phrase or not. According to the evaluation results, 86 collocations out of 247 were marked as true.

Further, we calculated the number of true collocations for each measure (within first 100 "cleaned" lines) (Table 4, the last line). The resulting number can be interpreted as the precision indicator (in percentage) for the upper part of the ranked list. This is also the indicator of the recall (or quasi-recall), i.e. how many collocations out of the potential 86 were obtained using this measure.

However, it is not only the number of the true collocations extracted using each measure that is important: the rank of the relevant collocation is significant, too. This is why it would be prudent to introduce a weight of true collocations for each measure taking into account the place of the collocates in a sorted table.

In order to evaluate the efficiency of each of the association measures the Kharin-Ashmanov method, which evaluates the relevance of the information retrieval results, was used [Ashmanov et al., 1997].

Based on the expert evaluation of the extracted collocates and their place in the ranked list with regard to each association measure, a characteristic set was formed. A characteristic set means the number of the true collocations obtained with different numbers of collocates from the ranked list (precision value).

According [Ashmanov et al., 1997], we select characteristic sets that contain 5 elements—the precision values for the first 10, 30, 50, 70 and 100 collocates from the top of the list (Table 4).

Table 4. Distribution of the number of true collocations for each measure

Ranks	T-score	MI	MI3	log likelihood	min. sensitivity	log-Dice	MI.log_f
1–10	0	5	4	2	5	6	8
1–30	6	10	11	9	11	9	15
1–50	8	18	18	14	13	15	23
1–70	10	28	21	22	14	19	26
1–100	17	39	33	26	21	22	31

A weight is assigned to each element of the characteristic set (5, 4, 3, 2, and 1, respectively). Each element is "weighed": each of 5 precision values is multiplied by the its weight and divided by 15 (the sum of the weights). The sum of the weighed elements is the resulting precision of the characteristic set.

Here is an example for the *MI* measure that has 5 true collocates in the top ten candidates (precision is 0.5), 10 true collocates in the top thirty (precision is 0.33), 18 in the top fifty (0.36), 28 in the top seventy (70), and 39 in the top hundred (0.39). Then, the resulting (average) precision will be equal to 0,5*5/15 + 0,33*4/15 + 0,36*3/15 + 0,4*2/15 + 0.39*1/15 = 0,167 + 0,088 + 0,072 + 0,053 + 0,026 = 0,406.

The values of the precision (let's call it *normalised precision*) calculated like that for all seven measures are given below (Table 5).

Table 5. Normalised precision values for association measures

	t-score	MI	MI3	log likelihood	min. sensitivity	log-Dice	MI.log_f
Number of true collocations	17	39	33	26	21	22	31
Normalised precision	0.115	**0.406**	0.366	0.262	0.357	**0.391**	**0.562**
Place	7	**2**	4	6	5	**3**	**1**

So, the best measure is *MI.l-og_f.* It is also shown that, for example, the precision of the *log-likelihood* measure (that has 26 true collocations) is lower than that of the *min. sensitivity* measure (that has 21 true collocation). The *MI* measure that has come second shall also be highlighted, since it is quite peculiar, despite its high performance. Here is the fragment from its collocate list (Table 5).

Table 6. Some collocates of *вода* (water) extracted by *MI* measure

	Co-occurrence count	Candidate count	MI
Омагниченная (magnetic)	6	5	11.581
Бахмут (bakhmut)	14	13	11.425
Бутилированная (bottled)	36	35	11.359
Мицеллярная (micellar)	10	10	11.318
Умягченная (soft)	5	5	11.318

This list can be continued: *деаэрируемая (deaerated), азотно-радоновая (nitrogen and radon), юрско-девонская (Jurassic & Devonian), подзоленная (ashen-gray), водородонасыщенная (reach in hydrogen).* On the one hand, this measure often extracts actual multi-word terms. On the other hand, it fails to identify or place into the "tail" of the list of well-known collocations. Erroneous spelling (*ки-пяченой, еесентуки*), proper names ("*Сент-Ронанские воды*"), nonce words, foreign words, words in Latin characters, etc. seem to be *MI* measure collocates, too. It should be noted that a lot of such words occur in large corpora. Of course such a "noise" in the corpus data can influence the results.

This is why we set a minimum limits for number of collocates in a corpus (the Sketch Engine has such parameters) to cut rare collocations, so that words with a frequency below a limit were not been included in the calculation.

New experiment with such a limitation and with other words gave the next results (Table 7, 8).

Table 7. Normalised precision values for association measures for *враг* (enemy) (with limitation of number of collocates)

	t-score	**MI**	**MI3**	**log likelihood**	**min. sensitivity**	**log-Dice**	**MI.log_f**
Number of true collocations	21	32	31	29	33	33	30
Normalised precision	0,266	0,505	0,362	0,373	**0,506**	**0,613**	**0,532**
Place	7	4	6	5	**3**	**1**	**2**

Table 8. Normalised precision values for association measures for *рыба* (fish) (with limitation of number of collocates)

	t-score	**MI**	**MI3**	**log likelihood**	**min. sensitivity**	**log-Dice**	**MI.log_f**
Number of true collocations	29	32	57	50	63	62	69
Normalised precision	0,229	0,340	0,572	0,495	**0,753**	**0,771**	**0,820**
Place	7	6	4	5	**3**	**2**	**1**

We see that MI measure in this case went down and that the best measures with respect to an expert evaluation are *MI.log_f, log-Dice* and *minimum sensitivity*. Similar results were obtained on the Russicum Russicum Maius corpus. So, we could conclude that efficiency of measures doesn't depend on corpus volume, at least, it is true for homogeneous web corpora.

Having obtained objective evaluation of the efficiency of individual measures, now we can introduce another rank indicator, which we will call the *optimised average rank*. This indicator is calculated taking into account the preference of the measures.

It is calculated as follows: all products of non-zero ranks multiplied by the coefficient of the measure significance are summed up and are divided into the number of measures used for a given collocate.

We suggest to set the measure significance coefficients, with their normalised precision taken into account (see Table 5, 7, 8 and also data from other experiments) as follows: *MI.log_f*—0.4, *logDice*—0.5, *min. sensitivity*—0.6, MI—0.7, MI3—0.8, *log-likelihood*—0.9, *T-score*—1.0. Of course, this is only preliminary ranking. The procedure of calculating normalised precision for association measures should be repeated on more words from different frequency belts.

As a result, the rank of the collocations extracted by more efficient measures is reduced, and the relevant collocate in the summary table goes up. See the example in Table 9.

Table 9. Optimised average rank for individual collocations

No.	Collocate	Average rank	Optimised average rank
1.	Поверхностный (surface)	**81.5**	59.8
2.	Крещенский (baptismal)	82.0	**36.0**
3.	Обычный (usual)	**61.0**	34.1
4.	Газированный (sparkling)	63.0	**27.9**
5.	Качество (quality)	**24.6**	19.5
6.	Урез (encroachment line)	29.0	**14.5**
7.	Соленый (salt)	**49.7**	37.2
8.	Паводковый (flood)	52.5	**22.5**

If you compare the collocates with even and odd numbers by pairs (*поверхностный* '*surface*' vs. *крещенский* '*baptismal*', *обычный* '*usual*' vs. *газированный* '*sparkling*', *качество* '*quality*' vs. *урез* '*encroachment line*', *соленый* '*salt*' vs. *паводковый* '*flood*'), then it is clear that the latter, having collocations with *water* as the node, still have a bit higher average rank than the former (which means that their average stability obtained through the integration of all the association measures is a bit lower). However, as per our suggestion, following the optimisation, the latter will have a lower rank and go up in the ranked list (once again: the higher the rank in this list the higher the collocability degree).

6. Conclusion

To sum it up, the experiments have produced important results that characterise the efficiency of individual association measures. It is important to stress that the experiments were conducted using representative corpora, with large amount of the resulting collocations being under study.

We offer a method of assessing the effectiveness of statistical association measures. The data on the normalised precision allows to establish to some degree that in cases when collocation extraction is not used for some special purposes such measures as *MI.l-og_f, log-Dice*, and *minimum sensitivity* should be used. The *MI* measure is critical when rare multi-word terms are needed to be extracted.

Conversely, no measure is ideal, which is why various options of their integration are desirable and useful. And we propose a number of parameters that allow to rank collocates in such combined list. Merging several lists of collocates obtained by different measures into one improves the efficiency of statistical tools in total. We offer several options that allow to assess "the quality" of collocations in the combined list.

7. Further work

Further research will be as follows:

1. Develop the programming tool that allows to make a single list of collocates with all the necessary parameters and calculate integrated ranks.

2. Study how the efficiency of the association measures is associated with the width of the range (to the left and to the right of the key word) within which collocates are selected, and estimate the degree of such efficiency.
3. Identify the inter-relation between "syntagmatic" and "paradigmatic" collocates on the one hand and "idiomatic" and "statistical" on the other hand within the same search results, and identify the dependence of such inter-relation on the width of the window.
4. Do research with data from dictionaries used as the gold standard.

Acknowledgement

This work was partly supported by the grant of the Russian Foundation for Humanities (research project No. 16-04-12019).

References

1. *Ashmanov I., Grigoryev S., Gusev V., Kharin N., Shabanov V.* (1997), Using Statistical Method for Intelligent Computer-Based Text Processing [Primenenie statisticheskih metodov dlja intellektual'noj komp'juternoj obrabotki tekstov] / The Proceedings of the Dialog'97 International Seminar on Computational Linguistics and Its Applications, pp. 33–37.
2. *Benko V.* (2014), Aranea: Yet another Family of (Comparable) Web Corpora, Text, Speech, and Dialogue. 17th International Conference, TSD 2014 Brno, Czech Republic, September 8–12, 2014, Proceedings, Ed. P. Sojka et al., pp. 247–256.
3. *Benko V., Zakharov V. P.* (2016), Very large Russian corpora : new opportunities and new challenges. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2016" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2006"], Vol. 15 (22). pp. 79–93.
4. *Braslavskij P., Sokolov J.* (2006), Comparison of four methods of automation extraction of two-word terms from text [Sravnenie četyreh metodov avtomatičeskogo izvlečenija dvuhslovnyh terminov iz teksta], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2006" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2006"], Bekasovo, pp. 88–94.
5. *Church K., Hanks P.* (1991), Word Association Norms, Mutual Information and Lexicography, Computational Linguistics, Vol 16:1, pp. 22–29.
6. *Daille B.* (1994), Mixed approach for the automatic extraction of terminology: lexical statistics and linguistic filters [Approche mixte pour l'extraction automatique de terminologie: statistiques lexicales et filtres linguistiques], PhD thesis, Université Paris 7.
7. *Dunning T.* (1993), Accurate Methods for the Statistics of Surprise and Coincidence, Computational Linguistics Vol. 19, Issue 1, pp. 61–74.

8. *Evert S., Krenn B.* (2001), Methods for the Qualitative Evaluation of Lexical Association Measures, ACL Proceedings of 39th Annual Meeting, Toulouse, France, pp. 188–195.
9. *Evert S.* (2004), The Statistics of Word Cooccurences Word Pairs and Collocations, PhD thesis. Institut für Maschinelle Sprachverarbeitung (IMS). Stuttgart.
10. *Halliday M.* (1991), Current Ideas in Systemic Practice and Theory. London.
11. *Kilgarriff A., Tugwell D.* (2002), Sketching words, Lexicography and Natural Language Processing: A Festschrift in Honour of B. T. S. Atkins, M. H. Correard (Ed.), Euralex, August, pp. 125–137.
12. *Kilgarriff A., Tugwell D.* (2001), WORD SKETCH: Extraction and Display of Significant Collocations for Lexicography, Proc. Workshop COLLOCATION: Computational Extraction, Analysis and Exploitation, 39th ACL & 10th EACL, Toulouse, France, pp. 32–38.
13. *Křen M.* (2006), Collocation Measures and the Czech Language: Comparison on the Czech National Corpus data [Kolokační míry a čeština: srovnání na datech Českého národního korpusu], Kolokace, Praha, pp. 223–248.
14. *Pecina P.* (2009), Lexical Association Measures. Collocation Extraction, Prague.
15. *Rychlý P.* (2007), Manatee/Bonito — A Modular Corpus Manager, 1st Workshop on Recent Advances in Slavonic Natural Language Processing, Brno, Masaryk University, pp. 65–70.
16. *Seretan V.* (2011), Syntax-based Collocation extraction. Text, Speech and Language, Springer Science.
17. *Sinclair J.* (1996), The Search for Units of Meaning, Textus, IX, pp. 75–106.
18. Statistics Used in Sketch Engine. URL: https://www.sketchengine.co.uk/documentation/statistics-used-in-sketch-engine/ (Last access 03.02.2017).
19. *Zakharov V., Khokhlova M.* (2014), Syntagmatic Relations in Russian Corpora and Dictionaries, Pragmantax II. The Present State of Linguistics and its Sub-Disciplines [Zum aktuellen Stand der Linguistik und ihren Teildisziplinen], Frankfurt a.M., Peter Lang, pp. 333–344.

PARAPHRASED PLAGIARISM DETECTION USING SENTENCE SIMILARITY

Zubarev D. V. (dvzubarev@yandex.ru),
Sochenkov I. V. (isochenkov@sci.pfu.edu.ru)

RUDN University, Moscow, Russia

Federal Research Center "Computer Science and Control"
of Russian Academy of Sciences, Moscow, Russia

The paper describes an approach to plagiarism detection within PlagEvalRus-2017 competition. Our system leverages deep parsing techniques to be able to detect moderately disguised plagiarism. We participated in the two tracks of the competition: source retrieval (sources detection) and text alignment (paraphrased plagiarism detection). There are various cases of plagiarism presented in datasets of both tracks. They vary by the level of disguise that was used while reusing text. The results show that our method performed quite well for detecting moderately disguised forms of plagiarism.

Keywords: plagiarism detection, sentence similarity, plagiarism detection evaluation

МЕТОД ПОИСКА ПЕРЕФРАЗИРОВАННЫХ ТЕКСТОВЫХ ЗАИМСТВОВАНИЙ НА ОСНОВЕ ОЦЕНКИ СХОДСТВА ПРЕДЛОЖЕНИЙ

Зубарев Д. В. (dvzubarev@yandex.ru),
Соченков И. В. (isochenkov@sci.pfu.edu.ru)

Российский университет дружбы народов, Москва, Россия

Федеральный исследовательский центр «Информатика
и управление» Российской академии наук, Москва, Россия

1. Introduction

Plagiarism is a serious and well-known problem in education and science. With the rapid increase of amount of texts available in digital form, it is crucial to detect sources of plagiarism fast enough in the huge number of documents.

Since plagiarism detection systems (PDS) are very common now, authors who reuse text have learned to disguise the fact of plagiarism. Since "copy-paste" plagiarism is likely to be detected, authors use various techniques such as paraphrasing, words reordering, split/join of sentences and so on (Alzahrani et al., 2012). Therefore, it is important for the state-of-the-art PDS to detect such cases also. Paraphrased plagiarism detection on a large amount of potential sources is a challenging task with no "gold-standard" solution for now. In general it is about to find a balance between false positives and false negatives in results of plagiarism detection. Therefore, it is important to evaluate information retrieval methods for plagiarism detection to find the most promising solutions.

PlagEvalRus-2017 is the first Russian competition for evaluation of plagiarism detection methods. It addresses the two main tasks in this area: source retrieval and text alignment. PlagEvalRus-2017 is a playground with the open dataset for researchers dealing with two aforementioned tasks. This dataset contains about 7 millions documents and plagiarism cases that vary by the level of complexity: from copy-paste plagiarism to heavily disguised plagiarism. In source retrieval track, participating PDS should find for given suspicious documents all sources in the entire collection. In text alignment track, the participants should detect all reused text between documents in the given pairs.

In this paper, we describe our approach for detecting plagiarism, which uses deep linguistic parsing of texts. It includes PoS-tagging, syntactic parsing, semantic role labeling, and semantic relation extraction. We also employ our own indexing subsystem that provides an efficient storage for rich information about words and an effective data access for the fast candidates' selection. The evaluation of our approach on the PlagEvalRus-2017 is also presented.

2. Related work

A comprehensive overview of approaches used to detect plagiarism is given in (Meuschke et al., 2013). Another overview along with taxonomy of plagiarism is given in (Alzahrani et al., 2012). Classic approach for detecting plagiarism is to use N-word-grams or N-character-grams. Recent research focused on incorporating syntactic and semantic information into detection methods. (Lin et al., 2012) used six similarity scores to measure the degree of plagiarism between fragments. Although they showed that impact of semantic and syntactic aspects to the overall performance was quite small. (Osman et al., 2012) measured sentence similarity based on semantic role labeling and achieved an improvement of more than 35% for both, precision and recall in comparison with classical methods.

It is very important to have standardized dataset, on which researchers can evaluate all new methods. An overview on the evaluation of plagiarism is given in (Kraus, 2016). Actually, there are few open datasets for such evaluation and mostly used is PAN-PC-11 corpus (Potthast et al., 2010). This corpus was used in PAN competition that held yearly since 2009 until 2015 year. The corpus consists of documents that were created by borrowing text of books from Gutenberg collection. Reused text was modified automatically and manually. Since the text is borrowed randomly from any book, the suspicious documents do not belong to the same topic as sources. This is the main concern related with this corpus and it makes it suitable only for evaluation of the text alignment task. Those corpora comprise documents are mostly in English.

3. The proposed Plagiarism detection method

In this section, we describe our method for external plagiarism detection. Our method relies on a collection of documents in which sources of a potentially plagiarized document would be located. Therefore an indexing subsystem is crucial for our method.

3.1. Data Indexing

We use our own indexing subsystem designed for an efficient (in terms of space on hard drive) storage of various words characteristics (PoS-tags, semantic roles etc.). To provide this information for indexing we perform linguistic analysis of texts, which includes postagging, syntactic parsing, semantic role labeling, and semantic relation extraction (Osipov et al., 2013), (Shelmanov and Smirnov, 2014). This information is used when we measure similarity between sentences.

3.2. Search method

One approach of searching text in the large collection is to use a search engine with special operators such as quorum matching or proximity search. This approach becomes impractical with a large number of documents. Each query consists of ten or more words and takes considerable time to complete. Since a search engine loads a list of occurrences of each word and then merges these lists into one to perform search. The amount of queries for a document depends on its size, but it is typically in order of hundreds. Our current method performs 18 times faster than our previous algorithm (based on search engines) for large documents (PhD theses) and 12 times faster for documents like medium wiki article.

Search of plagiarism is divided into three stages.

First stage

We represent the suspicious document as a bag of terms: words and two-word noun phrases. Each word and two-word noun phrase is normalized. These terms are sorted by the TF-IDF weight (IDF weights are calculated based on word and phrase frequencies in all indexed documents) and the top N terms with the highest weight are sent as a request to the indexing subsystem for retrieving similar documents. N is dependent on the amount of unique terms in a document (e.g. we select 45% of all terms but maximum is 120).

Indexing subsystem contains pre-built inverted spectral index for the whole collection of documents. This index stores a mapping from terms to their TF-IDF weights and a document (as the modification of the inverted index described in (Elsayed, 2008)). The index is employed for quick loading of all other vectors that overlap with the query vector. Then we calculate the modified Hamming distance to estimate the similarity score between suspicious document and documents in index. The full description of this method is given in (Suvorov and Sochenkov, 2015). We use 600 most similar documents as candidates: documents that may contain plagiarism. All other documents from the collection of sources are not taken into consideration.

Second stage

In this stage, we consider sentences as a sequence of words. We weight all sentences from the suspicious document by TF-IDF. The least significant ones (weight < 0.01) are dropped. In addition, we discard sentences that contain less than K or more than L words. We also discard all duplicate sentences. The remaining sentences will be analyzed for plagiarism.

We represent each sentence as a vector of unique numbers (each number is a derived from the normal form of corresponding word occurrence in sentence). Next, we intersect each selected sentence from the suspicious document with all other sentences from the candidates found on the previous stage. The goal is to exclude irrelevant pairs of sentences that share small amount of words from further consideration. For that task we use fast set intersection algorithm (Takuma, 2013). It proved to be very efficient for this task, since it boils down to multiple bitwise operations for each pair of sentence. Pairs of sentences that share at least *M%* of words are passed to the next stage.

Third stage

The remaining pairs of sentences are scored using a sentence similarity measure. We described this measure in (Zubarev and Sochenkov, 2014). We will only briefly recap it here.

Given two arbitrary sentences $s1$ and $s2$, denote as $N(s1, s2)$ a set of pairs of words with the same normal form, where the first element is taken from $s1$ and the second one from $s2$. We compare two sentences by considering words from the set $N(s1, s2)$. For calculating overall similarity measure of two sentences we compute multiple similarities measures and then combine its values. Employed similarities are described below.

IDF overlap measure

We define IDF overlap as follows:

$$I_1(s1, s2) = \sum_{(w1,w2)\in N(s1,s2)} v(w1, s1)$$

where $v(w1, s1)$ is IDF weight of word $w1$ in a sentence $s1$. Also there holds an equation

$$\sum_{w\in s1} v(w) = 1$$

TF-IDF measure

Let us define TF-IDF measure in the following way:

$$I_2(s1, s2) = \sum_{(w1,w2)\in N(s1,s2)} v(w1, s1) TF_{w2}$$

where $v(w1, s1)$ is IDF weight of the word $w1 \in s1$; TF_{w2} is TF weight of the word $w2 \in s2$.

Sentence syntactic similarity measure

To be able to measure this kind of similarity we need to use rich information stored in indexing subsystem for each word. We define *Syn*(*s*1) as a set that contains triplets (w_h, σ, w_d), where w_h, w_d are normalized head and dependent word respectively, σ is type of syntactic relation. Then we define syntactic similarity in the following way:

$$I_3(s1, s2) = \frac{\sum_{(w_h,\sigma,w_d)\in(Syn(s1)\cap Syn(s2))} v(w_h, s1)}{\sum_{(w_h,\sigma,w_d)\in Syn(s1)} v(w_h, s1)}$$

Sentence semantic similarity measure

For semantic information representation in a sentence we need to define:

- A finite set of semantic values—*Roles*. Further in the text we will call them roles (Shelmanov and Smirnov, 2014).
- *SentRoles*(*s*) is a set which contains pairs (*w*, ρ), where *w* is a normalized word from a sentence with an assigned role. Each word can have one or more semantic roles in the sentence.

Then we define semantic similarity in the following way:

$$I_4(s1, s2) = \frac{|SentRoles(s1) \cap SentRoles(s2)|}{|SentRoles(s1)|}$$

The denominator of the previous formula can be equal to 0 when no roles were identified in the sentence. In this case the criterion is equal to 0.

Sentence semantic relations similarity measure

For semantic relations representation in a sentence we need to define:

- Set of types of relations *R* on the set of semantic roles (Osipov et al., 2013).
- *SentRels*(*s*) is a set which contains pairs *w*1, *w*2, which determine semantically related words in a sentence, *w*1 and *w*2 should have any role assigned.

We define *SemR*(*s*, *w*) as a set

$$\{a \in Roles | \exists w1 \in s: (w, w1) \in SentRels(s) \wedge (w1, a) \in SentRoles(s)\}$$

It is a set of roles which were assigned to words *w*1 that are linked with words *w* in this sentence *s* by any semantic links. Then we define semantic relations similarity in the following way:

$$I_5(s1, s2) = \frac{\sum_{(w1,w2)\in N(s1,s2)} |SemR(s1, w1) \cap SemR(s2, w2)|}{|SentRoles(s1)|}$$

Overall sentence similarity

The overall sentence similarity we define as a linear combination of described measures.

$$Sim(s1, s2) = \sum_{i=1}^{5} k_i I_i(s1, s2),$$

where k_i, i = [1;5] determine relative contributions of each similarity.

Rationale for syntactic/semantic measures is to treat sentences not as a bag-of-words but as syntactically linked text with the meaning. Value of these measures will be low for sentences with the same words but with different usage of words.

Post-processing

There are two thresholds, which a pair of sentences must exceed to be considered as suspicious. First, a minimal value of IDF overlap measure and second, a minimal value of the overall sentence similarity.

Then all suspicious sentences are grouped by sources. Sources are sorted by the count of the sentences in them. We discard some sources: if they contain small number of sentences or if the percent of sentences from the total count is too small.

4. Tuning plagiarism detection method

There are many tunable parameters in the described method. We needed to tune 13 parameters each of them had from 10 to 20 values in general. It was not feasible to perform an exhaustive grid search for them. So we employed some kind of random search. At the beginning of search we initialize each parameter with a random value. Then we iterate over each parameter and tweak it by increasing/decreasing it slightly with respect of its bounds. On each iteration, we measure the performance of the detection method. The parameters from the best iteration are adopted as the current set of parameters and the search is repeated again. The search is interrupted, when the performance of the detection method is not changed for a while, and started again with new random values. We performed about 20 such restarts while optimizing parameters of the detection method. Mostly all searches converged to approximately one value with standard deviation 0.018. We optimized our method separately for text alignment and source retrieval tasks since these tasks use different performance measures.

5. Evaluation

5.1. Source retrieval task

We consider source retrieval as the first step of plagiarism detection, when all sources should be collected. Source retrieval occurs on the first stage in our plagiarism detection method. So we wanted to test how many sources we can find with our first stage.

Source retrieval training set includes plagiarism cases with various obfuscation types. **Academic** includes real world examples of plagiarism in academic environment (519 documents). This collection consists of PhD theses in which plagiarism was found. Texts from this collection contain copy-paste plagiarism in general.

Essays-1 (manually-paraphrased—name in the corpus) includes manually written essays on the given topic (118 documents). Authors of essays were asked to actively reuse other texts and change them. The texts from this collection may be described as being moderately disguised.

Essays-2 (manually-paraphrased2) the same as **Essays-1**, but they are heavily disguised in general (34 documents). **Generated plagiarism** includes suspicious documents generated automatically (1000 documents). We didn't evaluate this collection since the suspicious documents were filled with passages from the random sources and they are very likely on different topics. Hence it makes little sense trying to retrieve those sources. The results on the training dataset are presented in the following table.

Table 1. Results on the training data for source retrieval

	Recall	**Mean average precision**	**Precision**
Academic	0.97	0.359	0.001
Essays-1	0.983	0.149	0.009
Essays-2	0.969	0.118	0.009
Generated	—	—	—

The result shows that the first stage of our method is able to find most sources of plagiarism even when a search is performed against 7 millions documents. It means that most sources are in those 600 candidates that are left after the first stage and we still can find them in the next stages. Precision is low since we deliberately turn off any filtering of false candidates. We will show more balanced version of source retrieval when evaluating the whole method for detecting plagiarism in the next section.

Result on the test data were provided by the organizers. They are similar to results obtained on the training data, except that test data lacked Essays-1 collection.

Table 2. Results on the test data for source retrieval

	Recall	**Mean average precision**	**Precision**
Academic	0.978	0.61	0.003
Essays-2	0.989	0.39	0.009
Generated paraphrasing	0.75	0.2	0.005

5.2. Text alignment task

Text alignment is the crucial step of plagiarism detection, when reused text should be identified. Text alignment occurs on the second and third stage in our plagiarism detection method. For evaluating text alignment we use all stages except the first one, since a pair of documents is given in this task. Text alignment training set overlaps with the source retrieval training set. There is additional information in each corpus that is useful for text alignment task. In Essays-1 collection, authors annotated each pair of sentences with the type of obfuscation, which was used while modifying text. In **Essays-2**, authors were allowed to use more obfuscation types and each pair of sentences may be annotated with the multiple types. For example 'ADD,SYN' means that there were used addition of words and replacing some words with synonyms.

Standard metrics for text alignment were used to evaluate our approach:

- micro-averaged recall and precision;
- granularity is used to penalty multiple detections for a single plagiarism case (the higher the worse);
- plagdet—the overall score that is a combination of the previous three measures.

More information about these metrics can be found in (Potthast et al., 2010). Results obtained on the training data are shown in the next table.

Table 3. Results on the training data for text alignment

	Recall	Precision	Granularity	Plagdet
Essays-1	0.848	0.862	1.0011	0.854
Essays-2	0.463	0.824	1.0026	0.591
Generated copy/paste	0.756	0.977	1.41	0.672
Generated paraphrasing	0.706	0.982	1.53	0.614

We can see strong decrease of recall when difficulty of obfuscations is increased for both generated texts and manually written. Also it is clear that our method does not find all cases even for moderately disguised plagiarism. Low recall for generated plagiarism cases is rather surprising. The cause may be that the generated suspicious documents contain duplicate sentences taken multiple times from a single source file. We discard all duplicate sentences in the second stage of our method. Therefore, we can't find all of them.

Since the training data was annotated with the type of obfuscation used when modifying each fragment, we were able to identify the most difficult types of obfuscation for our method. The result for the collection **Essays-1** is presented below.

Table 4. Recall per obfuscation type

	Description	Recall
CCT	concatenation of sentences	0.41
HPR	paraphrasing	0.44
SSP	splitting of sentences	0.65
LPR	moderate modifications (replacing/reordering of words)	0.78
ADD	addition of words	0.85
DEL	deletion of words	0.85
CPY	copy/paste	0.87

This result shows that the most difficult type of obfuscation for our method is concatenation of sentences and paraphrasing. The latter is quite understandable but the former is the limitation of our sentence based approach. The most of such fragments are lost in the third stage, since sentences from a source failed to provide sufficient IDF-overlap. The distribution of recall is similar for the collection **Essays-2**.

Result on the test data were provided by the organizers. They also provided a comparison with the baseline on the same collections.

Table 5. Results on the test data for text alignment

	Recall	Precision	Granularity	Plagdet
Essays-2	0.531	0.82	1.0016	0.644
Baseline: Essays-2	0.076	0.896	1.141	0.128
Generated paraphrasing	0.865	0.981	1.483	0.7
Baseline: generated paraphrasing	0.833	0.97	3.464	0.416
Generated copy/paste	0.859	0.978	1.466	0.702
Baseline: generated copy/paste	0.994	0.961	1.004	0.9744

Our method is better than baseline for all collections except the generated copy/paste collection.

5.3. Evaluation of plagiarism detection method

For evaluating all stages of our method at once, we use collections **Essays-2** and **Essays-1**, since we evaluated on both subtasks. We performed optimization on **Essays-2** collection with the goal to maximize Mean Average Precision (MAP).

Results on the training data are shown in the next table.

Table 6. Results on the training data for the whole method

	Source Retrieval			Text Alignment			
	Recall	Mean average precision	Precision	Recall	Precision	Granularity	Plagdet
Essays-1	0.97	0.754	0.332	0.783	0.904	1.00089	0.839
Essays-2	0.82	0.709	0.652	0.316	0.883	1.00095	0.466

This result shows that our method returns most sources in the top of the search results, since MAP is high relative to precision. It detects about of 80% of moderately disguised text and only a third of the text that was heavily paraphrased.

Similar results were obtained for test data, provided by organizers.

Table 7. Results on the test data for the whole method

	Source Retrieval			Text Alignment			
	Recall	Mean average precision	Precision	Recall	Precision	Granularity	Plagdet
Essays-2	0.83	0.608	0.441	0.382	0.885	1.0015	0.533

6. Conclusion

In this paper, we described our method for plagiarism detection and evaluation of this method in two tracks of PlagEvalRus-2017. The method was performed quite well for various plagiarism cases. The best result was achieved for manually written essays with moderately disguised plagiarism. PlagEvalRus corpus helped to identify some weak points of our method, which we are going to address in future. We also plan to estimate current impact of semantic/syntactic similarity measures on recall, and explore more possibilities to leverage them for detecting heavily disguised plagiarism.

References

1. *Alzahrani S. M., Salim N., Abraham A.* (2012), Understanding plagiarism linguistic patterns, textual features, and detection methods, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42(2), pp. 133–149.
2. *Elsayed T., Lin J., Oard D. W.* (2008), Pairwise document similarity in large collections with MapReduce, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 265–268.
3. *Kraus K.* (2016), Plagiarism Detection—State-of-the-art systems (2016) and evaluation methods, available at: http://arxiv.org/abs/1603.03014
4. *Lin W. Y., Peng N., Yen C. C., Lin, S. D.* (2012), Online plagiarism detection through exploiting lexical, syntactic, and semantic information, Proceedings of the ACL 2012 System Demonstrations, pp. 145–150.
5. *Meuschke N., Gipp B.* (2013), State-of-the-art in detecting academic plagiarism, International Journal for Educational Integrity, vol. 9(1), pp. 50–71
6. *Osipov G., Smirnov I., Tikhomirov I., Shelmanov A.* (2013), Relational-situational method for intelligent search and analysis of scientific publications, Proceedings of the Integrating IR Technologies for Professional Search Workshop, pp. 57–64.
7. *Osman A. H., Salim N., Binwahlan M. S., Alteeb R., Abuobieda A.* (2012), An improved plagiarism detection scheme based on semantic role labeling, Applied Soft Computing, vol. 12(5), pp. 1493–1502.
8. *Potthast M., Stein B., Barrón-Cedeño A., Rosso P.* (2010). An evaluation framework for plagiarism detection, Proceedings of the 23rd international conference on computational linguistics: Posters, Beijing, pp. 997–1005.
9. *Shelmanov A. O., Smirnov I. V.* (2014) Methods for semantic role labeling of Russian texts, Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" No. 13, pp. 607–620.
10. *Suvorov R. E., Sochenkov I. V.* (2015). Establishing the similarity of scientific and technical documents based on thematic significance, Scientific and Technical Information Processing, vol. 42(5), pp. 321–327.
11. *Takuma D., Yanagisawa H.* (2013), Faster upper bounding of intersection sizes, Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 703–712
12. *Zubarev D., Sochenkov I.* (2014), Using Sentence Similarity Measure for Plagiarism Source Retrieval, In CLEF (Working Notes), pp. 1027–1034.

Abstracts

PART-OF-SPEECH TAGGING WITH RICH LANGUAGE DESCRIPTION

Anastasyev D. G. (daniil_an@abbyy.com), **Andrianov A. I.** (andrew_an@abbyy.com), **Indenbom E. M.** (eugene_i@abbyy.com), ABBYY, Moscow, Russia

This paper deals with morphological parsing of natural language texts. We propose a method that combines comprehensive morphological description provided by ABBYY Compreno system and sophisticated machine learning techniques used by the state-of-the-art POS taggers. The morphological description contains information about possible grammatical values of a dictionary word that helps to identify a set of potential hypothesis for each word during the morphological analysis stage. To analyse out-of-vocabulary words we are building a number of most likely paradigms in the morphological model using the orthographic features of the analysed word. The proposed method helps to reduce the number of hypotheses using the context information of each word. We use Bidirectional LSTM classifier to handle the context information and to predict the most probable grammatical value. The ambiguous grammatical values obtained from morphological description are used as features for the classifier. Also, we use word embeddings and orthographic features to achieve better results.

SEMANTIC DESCRIPTIONS FOR A TEXT UNDERSTANDING SYSTEM

Boguslavsky I. (bogus@iitp.ru), Laboratory of Computational Linguistics, A. A. Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Russia; Departamento de Inteligencia Artificial, ETSII, Universidad Politécnica de Madrid, Spain

The semantic analyser SemETAP is a module of the ETAP-3 Linguistic Processor. It uses 2 static semantic resources—the combinatorial dictionary and the ontology. The former contains multifarious information about the words, and the latter stores extralinguistic (world) knowledge on the concepts and serves as the metalanguage for semantic description. World knowledge is needed, on the one hand, to enhance text analysis, and, on the other hand, to extract implicit information by means of inference. Both words and concepts are supplied with semantic descriptions. A semantic description consists of a definition in a formal language, which can optionally contain implications and expectations. For user's convenience, the description may also be provided by examples and a definition in NL. Semantic descriptions of several words and concepts are given.

WHICH IR MODEL HAS A BETTER SENSE OF HUMOR? SEARCH OVER A LARGE COLLECTION OF JOKES

Bolotova V. V. (lurunchik@gmail.com), **Blinov V. A.** (vladislav.blinov@urfu.ru), **Mishchenko K. I.** (ki.mishchenko@gmail.com), **Braslavski P. I.** (pbras@yandex.ru), Ural Federal University, Yekaterinburg, Russia

This paper describes experiments on humorous response generation for short text conversations. Firstly, we compiled a collection of 63,000 jokes from online social networks (VK and Twitter). Secondly, we implemented several context-aware joke retrieval models: BM25 as a baseline, query term reweighting, word2vec-based model, and learning-to-rank approach with multiple features. Finally, we evaluated these models in two ways: on the community question answering platform Otvety@Mail.ru and in laboratory settings. Evaluation shows that an information retrieval approach to humorous response generation yields satisfactory performance.

TEXT NORMALIZATION IN RUSSIAN TEXT-TO-SPEECH SYNTHESIS: TAXONOMY AND PROCESSING OF NON-STANDARD WORDS

Cherepanova O. D. (cherepanova.od@gmail.com), Moscow State University, Moscow, Russia

Alongside with ordinary words, natural-language text also contains non-standard words (NSWs), such as abbreviations, acronyms, dates, phone numbers, currency amounts etc. Before phonetizing these text elements in Text-to-Speech synthesis, it is necessary to normalize them by replacing them with an appropriate ordinary word or word sequence. NSWs are increasingly diverse and most of them require specific normalization rules. In this paper, we present a taxonomy of NSWs for the Russian language developed on the basis of news texts, software and car reviews and instruction manuals. We grouped NSWs that have similar normalization rules or patterns taking into account their graphic form and their context dependence. We propose five main groups of NSWs: abbreviations (including acronyms and initialisms), text elements containing numbers, special characters, foreign words written in the Latin alphabet and mixed-type non-standard words. In this work, we describe these NSW types and address the issue of their normalization in Russian Text-to-Speech synthesis.

RUSSIAN COLLOCATION EXTRACTION BASED ON WORD EMBEDDINGS

Enikeeva E. V. (protoev@yandex.ru), **Mitrofanova O. A.** (o.mitrofanova@spbu.ru),
Saint Petersburg State University, St. Petersburg, Russia

Collocation acquisition is a crucial task in language learning as well as in natural language processing. Semantics-oriented computational approaches to collocations are quite rare, especially on Russian language data, and require an underlying semantic formalism. In this paper we exploit a definition of collocation by I. A. Mel'čuk and colleagues (Iordanskaya, Mel'čuk 2007) and apply the theory of lexical functions to the task of collocation extraction. Distributed word vector models serve as a state-of-the-art computational basis for the tested method. For the first time experiments of such type are conducted on available Russian language data, including Russian National Corpus, SynTagRus and RusVectōrēs project resources. The resulting collocation lists are assessed manually and then evaluated by means of precision and MRR metrics. Final scores are quite promising (reaching 0.9 in precision) and described algorithm improvements yield a considerable performance growth.

COMPARATIVE ANALYSIS OF ANGLICISM DISTRIBUTION IN RUSSIAN SOCIAL NETWORK TEXTS

Fenogenova A. S. (alenush93@gmail.com), **Karpov I. A.** (karpovilia@gmail.com),
Kazorin V. I. (zhelyazik@mail.ru), **Lebedev I. V.** (innlebedev@gmail.com),
National Research University Higher School of Economics, Research and Development Institute KVANT, Moscow, Russia

Due to the process of globalization, the number of English borrowings in different languages is constantly growing. In natural language processing (NLP) systems, such as spell-check, POS tags, etc. the analysis of loan words is not a trivial task and should be resolved separately. This article continues our previous work on the corpus-driven Anglicism detection by proposing an improved method to the search of loan words by means of contemporary machine translation methods. It then describes distribution of the borrowed lexicon in different online social networks (OSN) and blog platforms showing that the Anglicism search task strongly depends on corpus formation method. Our approach does not contain any pre-prepared, manually acquired data and gives a significant automation in Anglicism dictionary generation. We present an effective dictionary collection method that gives the same coverage compared to random user selection strategy on a 20 times smaller corpus. Our comparative study on LiveJournal, VKontakte, Habrahabr and Twitter shows that different social, gender, even age groups have the same proportion of Anglicisms in speech.

LEARNING NOISY DISCOURSE TREES

Galitsky B. (boris.galitsky@oracle.com), Oracle Corp Redwood Shores CA USA

It is well known that syntax-level analysis of user-generated text such as tweets and forum postings is unreliable due to its poor grammar and incompleteness. We attempt to apply a higher level linguistic analysis of rhetoric structure and investigate the potential application domains. We leverage an observation that discourse-level structure can be extracted from noisy text with higher reliability than syntactic links and named entities. As noisy text frequently includes informal interaction between agents, discussions, negotiations, arguments, complaints, we augment discourse trees with speech acts. Speech Act discourse tree (SADT) is defined as a discourse tree with verbs for speech acts as labels for its arcs. We identify text classification tasks which relies on tree kernel learning of SADTs: detection of negative mood (sentiment), text authenticity and answer appropriateness for question answering in social domains. The results are that the proposed technique outperforms on the discourse level traditional keyword-based algorithms in all of these three tasks.

COMPLEX APPROACH TOWARDS ALGORITM LEARNING FOR ANAPHORA RESOLUTION IN RUSSIAN LANGUAGE

Gureenkova O. A. (ol.gure@gmail.com)[1], **Batura T. V.** (tatiana.v.batura@gmail.com)[2,3], **Kozlova A. A.** (noriel266@gmail.com)[2], **Svischev A. N.** (alekseisvischev@gmail.com)[2]
[1]Expasoft Ltd., Novosibirsk, Russia; [2]Novosibirsk State University, Novosibirsk, Russia;
[3]A. P. Ershov Institute of Informatics systems, Novosibirsk, Russia

The paper considers applying of ensemble algorithm based on rules and machine learning for anaphora resolution in Russian language. Ensemble presents combination of formal rules, a machine learning algorithm Extra Trees and an algorithm for working with imbalanced learning sets Balance Cascade. Complexity of the approach lies in generation of complex features from rules and vectorization of syntactic context, with context data obtained from algorithms mystem (Yandex), SyntaxNet (Google) and Word2Vec.

PART-OF-SPEECH TAGGING: THE POWER OF THE LINEAR SVM-BASED FILTRATION METHOD FOR RUSSIAN LANGUAGE

Kazennikov A. O. (kazennikov@iqmen.ru), IQMen LLC, Moscow, Russia

We present our approach to Part-of-Speech tagging and lemmatization tasks for Russian language in the context of MorphoRuEval-2017 Shared Task. The approach ranked second on the closed track and on several test subsets it ranked first.

We proposed a filtration-based method which seamlessly integrates a classical morphological analyzer approach with machine learning based filtering. The method addresses both tasks in a unified fashion. Our method consists of two stages. On the first stage we generate a set of candidate substitutions which simultaneously recovers the normal form and provides all necessary morphological information. We select an optimal substitution for the current word given its context on the second stage.

The filtration stage of the presented method is based on Linear SVMs extended with hash kernel. The extension reduces the size of our model by an order of magnitude and allows to easily tune the tradeoff between the precision and the model size.

ARBITRARINESS OF LINGUISTIC SIGN QUESTIONED: CORRELATION BETWEEN WORD FORM AND MEANING IN RUSSIAN

Kutuzov A. B. (andreku@ifi.uio.no), University of Oslo, Norway

In this paper, we present the results of preliminary experiments on finding the link between the surface forms of Russian nouns (as represented by their graphic forms) and their meanings (as represented by vectors in a distributional model trained on the Russian National Corpus). We show that there is a strongly significant correlation between these two sides of a linguistic sign (in our case, word). This correlation coefficient is equal to 0.03 as calculated on a set of 1,729 mono-syllabic nouns, and in some subsets of words starting with particular two-letter sequences the correlation raises as high as 0.57. The overall correlation value is higher than the one reported in similar experiments for English (0.016).

Additionally, we report correlation values for the noun subsets related to different phonaesthemes, supposedly represented by the initial characters of these nouns.

WORD SENSE INDUCTION FOR RUSSIAN: DEEP STUDY AND COMPARISON WITH DICTIONARIES

Lopukhin K. A. (kostia.lopuhin@gmail.com), Scrapinghub;
Iomdin B. L. (iomdin@ruslang.ru), V. V. Vinogradov Russian Language Institute of the Russian Academy of Sciences, National Research University Higher School of Economics;
Lopukhina A. A. (alopukhina@hse.ru), National Research University Higher School of Economics, V. V. Vinogradov Russian Language Institute of the Russian Academy of Sciences

The assumption that senses are mutually disjoint and have clear boundaries has been drawn into doubt by several linguists and psychologists. The problem of word sense granularity is widely discussed both in lexicographic and in NLP studies. We aim to study word senses in the wild—in raw corpora—by performing word sense induction (WSI). WSI is the task of automatically inducing the different senses of a given word in the form of an unsupervised learning task with senses represented as clusters of token instances. In this paper, we compared four WSI techniques: Adaptive Skip-gram (AdaGram), Latent Dirichlet Allocation (LDA), clustering of contexts and clustering of synonyms. We quantitatively and qualitatively evaluated them and performed a deep study of the AdaGram method comparing AdaGram clusters for 126 words (nouns, adjectives, and verbs) and their senses in published dictionaries. We found out that AdaGram is quite good at distinguishing homonyms and metaphoric meanings. It ignores disappearing and obsolete senses, but induces new and domain-specific senses which are sometimes absent in dictionaries. However it works better for nouns than for verbs, ignoring the structural differences (e.g. causative meanings or different government patterns). The Adagram database is available online: http://adagram.ll-cl.org/.

TESTING FEATURES AND METHODS IN RUSSIAN PARAPHRASING TASK

Loukachevitch N. V. (louk_nat@mail.ru), **Shevelev A. S.** (alex.shevelev@hotmail.com), **Mozharova V. A.** (joinmek@rambler.ru), Lomonosov Moscow State University, Moscow, Russia

In this paper we study several groups of features and machine learning methods in the shared task on Russian paraphrasing organized in 2016. We use four groups of features: string-based features, information-retrieval features, part-of-speech features and thesaurus-based features and compare three machine learning methods: SVM with linear and RBF kernels, Random Forest and Gradient Boosting. In our experiments, the best results were obtained with the Random Forest classifier with parameter tuning and using all groups of features. The results of Gradient Boosting with parameter tuning were slightly worse.

DOMAIN-INDEPENDENT CLASSIFICATION OF AUTOMATIC SPEECH RECOGNITION TEXTS

Mescheryakova E. I. (e-meshch@yandex.ru), **Nesterenko L. V.** (lyu.klimenchenko@gmail.com), National Research University Higher School of Economics; DC-Systems, Moscow, Russia

Call centers receive large amounts of incoming calls. The calls are being regularly processed by the analytical system, which helps people automatically inspect all the data. Such system demands a classification module that can determine the topic of conversation for each call. Due to high costs of manual annotation, the input for this module is the automatically transcribed calls. Hence, the texts (=automatic transcription) used for classification contain ill-transcribed words which can probably influence the classification process. Another important point is that this module also has special requirements: it should be domain-independent and easy to setup. Document classification task always requires an annotated data set for classifier training, but it seems to be too costly to make an annotated training set for each domain manually. In this paper, we propose an approach to automatic speech recognition texts classification that allows the user avoiding full manual annotation and at the same time to control its quality.

IDENTIFYING DISEASE-RELATED EXPRESSIONS IN REVIEWS USING CONDITIONAL RANDOM FIELDS

Miftahutdinov Z. Sh. (zulfatmi@gmail.com)[1], **Tutubalina E. V.** (elvtutubalina@kpfu.ru)[1], **Tropsha A. E.** (alex_tropsha@unc.edu)[1,2]
[1]Kazan Federal University, Kazan, Russia [2]University of North Carolina, Chapel Hill, USA

As the as the volume of user-generated content in social media expands so do the potential benefits of mining social media to learn about patient conditions, drug indications, and beneficial or adverse drug reactions. In this paper, we apply Conditional Random Fields (CRF) model for extracting expressions related to diseases from patient comments. Our method utilizes handcrafted features including contextual features, dictionaries, cluster-based and distributed word representation generated from unlabeled user posts in social media. We compare our CRF-based approach with deep recurrent neural networks and a dictionary-based approach. We examine different word embeddings generated from unlabeled user posts in social media and scientific literature. We show that CRF outperformed other methods and achieved the F_1-measures of 69.1% and 79.4% on recognition of disease-related expressions in the exact and partial matching exercises, respectively. Qualitative evaluation of disease-related expressions recognized by our feature-rich CRF-based approach demonstrates the variability of reactions from patients with different health conditions.

DETECTING INTENTIONAL LEXICAL AMBIGUITY IN ENGLISH PUNS

Mikhalkova E. V. (e.v.mikhalkova@utmn.ru), **Karyakin Yu. E.** (y.e.karyakin@utmn.ru), Tyumen State University, Tyumen, Russia

The article describes a model of automatic analysis of puns, where a word is intentionally used in two meanings at the same time (the target word). We employ Roget's Thesaurus to discover two groups of words, which, in a pun, form around two abstract bits of meaning (semes). They become a semantic vector, based on which an SVM classifier learns to recognize puns, reaching a score 0.73 for F-measure. We apply several rule-based methods to locate intentionally ambiguous (target) words, based on structural and semantic criteria. It appears that the structural criterion is more effective, although it possibly characterizes only the tested dataset. The results we get correlate with the results of other teams at SemEval-2017 competition (Task 7 Detection and Interpretation of English Puns), considering effects of using supervised learning models and word statistics.

DISTRIBUTIONAL SEMANTIC FEATURES IN RUSSIAN VERBAL METAPHOR IDENTIFICATION

Panicheva P. V. (ppolin86@gmail.com), **Badryzlova Yu. G.** (yuliya.badryzlova@gmail.com), St. Petersburg State University, Saint Petersburg, Russia; National Research University Higher School of Economics (HSE), Moscow, Russia

Our experiment is aimed at evaluating the performance of distributional semantic features in metaphor identification in Russian raw text. We apply two types of distributional features representing similarity between the metaphoric/literal verb and its syntactic or linear context. Our approach is evaluated on a dataset of nine Russian verb context, which is made available to the community. The results show that both sets of similarity features are useful for metaphor identification, and do not replicate each other, as their combination systematically improves the performance for individual verb sense classification, reaching state-of-the-art results for verbal metaphor identification. A combined verb classification demonstrates that the suggested features effectively generalize over metaphoric usage in different verbs, shows that linear coherence features perform as well as the combined feature approach. By analyzing the errors we conclude that syntactic parsing quality is still modest for raw-text metaphor identification in Russian, and discuss properties of semantic models required for high performance.

RHETORICAL STRUCTURE THEORY AS A FEATURE FOR DECEPTION DETECTION IN NEWS REPORTS IN THE RUSSIAN LANGUAGE

Pisarevskaya D. (dinabpr@gmail.com), Institute for System Programming of the RAS, Moscow, Russia

The framework of the Rhetorical Structure Theory (RST) can be used to reveal the differences between structures of truthful and deceptive (fake) news. This approach was already used for English. In this paper it is applied to Russian. Corpus consists of 134 truthful and deceptive news stories in Russian. Texts annotations contain 33 relation categories. Three data sets of experimental data were created: with only rhetorical relation categories (frequencies), with rhetorical relation categories and bigrams of categories, with rhetorical relation categories and trigrams of categories. Support Vector Machines and Random Forest Classifier were used for text classification. The best results we got by using Support Vector Machines with linear kernel for the first data set (0.65). The model could be used as a preliminary filter for fake news detection.

TOWARDS BUILDING A DISCOURSE-ANNOTATED CORPUS OF RUSSIAN

Pisarevskaya D. (dinabpr@gmail.com)[1], **Ananyeva M.** (ananyeva@isa.ru)[2], **Kobozeva M.** (kobozeva@isa.ru)[2], **Nasedkin A.** (kloudsnuff@gmail.com)[3], **Nikiforova S.** (son.nik@mail.ru)[3], **Pavlova I.** (ispavlovais@gmail.com)[3], **Shelepov A.** (alexshelepov1992@gmail.com)[3]

[1]Institute for System Programming of the RAS, Moscow, Russia; [2]Institute for Systems Analysis FRC CSC RAS, Moscow, Russia; [3]NRU Higher School of Economics, Moscow, Russia

For many natural language processing tasks (machine translation evaluation, anaphora resolution, information retrieval, etc.) a corpus of texts annotated for discourse structure is essential. As for now, there are no such corpora of written Russian, which stands in the way of developing a range of applications. This paper presents the first steps of constructing a Rhetorical Structure Corpus of the Russian language. Main annotation principles are discussed, as well as the problems that arise and the ways to solve them. Since annotation consistency is often an issue when texts are manually annotated for something as subjective as discourse structure, we specifically focus on the subject of inter-annotator agreement measurement. We also propose a new set of rhetorical relations (modified from the classic Mann & Thompson set), which is more suitable for Russian. We aim to use the corpus for experiments on discourse parsing and believe that the corpus will be of great help to other researchers. The corpus will be made available for public use.

BRIDGING ANAPHORA RESOLUTION FOR THE RUSSIAN LANGUAGE

Roitberg A. M. (cvi@yandex.ru)[1,2], **Khachko D. V.** (mordol@lpm.org.ru)[1]

[1]IMPB RAS- Branch of KIAM RAS, Puschino, Russia; [2]School of Linguistics HSE RSU, Moscow, Russia

Presented in this report are the initial findings of automatic bridging anaphora recognition and resolution for the Russian language. For a resolution of F-measure = 0.65 we use a manually-annotated bridging corpus and machine-learning techniques to develop a classifier to predict bridging anaphors, bridging anchors, and bridging pairs. In addition to this, we discuss the features used for the classifier and discuss the importance of each feature. Experimental results show that our classifier works well, however, potential improvements can be made, these improvements will be explored.

EXPLOITING RUSSIAN WORD EMBEDDINGS FOR AUTOMATED GRAMMEME PREDICTION

Romanov A. V. (Aleksey_Ro@abbyy.com), ABBYY; Moscow Institute of Physics and Technology (MIPT), Moscow, Russia

Distributed representations of words are currently used in a variety of linguistic tasks. A specific branch of their possible applications includes automatic extraction of word-level grammatical information by formulating it as a problem of word embedding classification. In this paper, we investigate applicability of this approach to prediction of several particular classifying grammemes. We focus on animacy of Russian nouns and transitivity of Russian verbs. These categories can serve as good examples of classifying grammatical categories in the Russian language since their concrete values can hardly be predicted judging by appearance of words and morphemes that constitute them. We conduct experiments on a corpus of Russian texts from the Web with several widely used word-embedding algorithms and different parameter settings. Experimental evaluation includes the comparison of performance of several classifiers, with distributed representations being source of features for classification task. Our findings show feasibility of the approach and its potential to be implemented for solving related tasks.

RESEARCH OF A DEEP LEARNING NEURAL NETWORK EFFECTIVENESS FOR A MORPHOLOGICAL PARSER OF RUSSIAN LANGUAGE

Sboev A. G. (sag111@mail.ru)[1,2,3],**Gudovskikh D. V.** (dvgudovskikh@gmail.com)[1],
Ivanov I. (honala@yandex.ru)[3], **Moloshnikov I. A.** (ivan-rus@yandex.ru)[1],
Rybka R. B. (rybkarb@gmail.com)[1], **Voronina I.** (irina.voronina@gmail.com)[4]
[1]National Research Center «Kurchatov Institute», Moscow, Russia; [2]National Research Nuclear University «MEPhI», Moscow, Russia; [3]Moscow Technological University (MIREA), Moscow, Russia; [4]Voronezh State University, Voronezh, Russian Federation

In this study we present the method of morphological tagging on base of a deep learning neural network. The method includes two levels of an input sentence processing: individual characters level and word level. The comparison with other morphological analyzers was carried out with SynTagRus dataset in its original format of morphological characters, and its versions in Universal Dependencies formats 1.3 and 1.4. Achieved accuracies of Part-of-speech tagging: 98.34%, 98.49%, 97.60% (accordingly to each dataset). Results are a bit higher than the Google Syntaxnet accuracies and higher than the accuracies of the systems based only on Bidirectional Long short-term memory models. At the MorphoRuEval competition the method gained the third place.

SEMANTIC ROLE LABELING WITH NEURAL NETWORKS FOR TEXTS IN RUSSIAN

Shelmanov A. O. (shelmanov@isa.ru), **Devyatkin D. A.** (devyatkin@isa.ru),
Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

We present and evaluate neural network models for semantic role labeling of texts in Russian. The benchmark for evaluation and training was prepared on the basis of the FrameBank corpus. The paper addresses different aspects of learning a neural network model for semantic role labeling on different feature sets including syntactic features acquired with the help of SyntaxNet. In this work, we rely on architecture engineering and atomic features instead of commonly used feature engineering. We investigate the ability of learning a model for labeling arguments of "unknown" predicates that are not present in a training set using word embeddings as features for the replacement of predicate lemmas. We publish the prepared benchmark and the models. The experimental results can be used as a baseline for further research in semantic role labeling of texts in Russian.

EXTRACTING CHARACTER NETWORKS TO EXPLORE LITERARY PLOT DYNAMICS

Skorinkin D. A. (dskorinkin@hse.ru), Higher School of Economics, Moscow, Russia

In this paper we apply network analysis to the study of literature. At the first stage of our investigation we automatically extract networks (graphs) of characters for each part of Leo Tolstoy's novel *War and peace* using two different techniques for network creation. Then we evaluate these two techniques against a set of manually created gold standard networks. Finally, we use the method that demonstrated better performance in our evaluation to test a literary hypothesis about Tolstoy's novel. The hypotheses we intended to prove was that the parts of the novel describing war (i.e. those where the battlefield or military units are the primary settings), have statistically lower density of interaction between characters, resulting in lower network den-

sity, higher network diameters and lesser average node degrees. By showing this correlation we mean to demonstrate the applicability of network analysis to computational research of fictional narrative (e.g. detection of tension changes in the plot).

EVALUATION TRACKS ON PLAGIARISM DETECTION ALGORITHMS FOR THE RUSSIAN LANGUAGE

Smirnov I. (ivs@isa.ru), Institute for Systems Analysis, FRC CSC RAS, Moscow, Russia; RUDN University, Moscow, Russia; **Kuznetsova R.** (kuznetsova@ap-team.ru), Antiplagiat JSC, Moscow, Russia; **Kopotev M.** (mihail.kopotev@helsinki.fi), University of Helsinki, Helsinki, Finland; **Khazov A.** (hazov@ap-team.ru), Antiplagiat JSC, Moscow, Russia; **Lyashevskaya O.** (olesar@yandex.ru), Higher School of Economics, Moscow, Russia; Vinogradov Institute of the Russian Language RAS, Moscow, Russia; **Ivanova L.** (luben92@gmail.com), Higher School of Economics, Moscow, Russia; **Kutuzov A.** (andreku@ifi.uio.no), University of Oslo, Oslo, Norway

The paper presents a methodology and preliminary results for evaluating plagiarism detection algorithms for the Russian language. We describe the goals and tasks of the PlagEvalRus workshop, dataset creation, evaluation setup, metrics, and results.

THE PARAPLAG: RUSSIAN DATASET FOR PARAPHRASED PLAGIARISM DETECTION

Sochenkov I. V. (sochenkov_iv@rudn.university)[1,2], **Zubarev D. V.** (zubarev@isa.ru)[1,2], **Smirnov I. V.** (ivs@isa.ru)[3,1]

[1]RUDN University, Moscow, Russia; [2]Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia; [3]Institute for Systems Analysis, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

The paper presents the ParaPlag: a large text dataset in Russian to evaluate and compare quality metrics of different plagiarism detection approaches that deal with big data. The competition PlagEvalRus-2017 aimed to evaluate plagiarism detection methods uses the ParaPlag as a main dataset for source retrieval and text alignment tasks. The ParaPlag is open and available on the Web. We propose a guide for writers who want to contribute to the ParaPlag and extend it. The analysis of text rewrite techniques used by unscrupulous authors is also presented in our research.

MORPHORUEVAL-2017: AN EVALUATION TRACK FOR THE AUTOMATIC MORPHOLOGICAL ANALYSIS METHODS FOR RUSSIAN

Sorokin A. (alexey.sorokin@list.ru)[1,2,6],**Shavrina T.** (rybolos@gmail.com)[4,6], **Lyashevskaya O.** (olesar@yandex.ru)[4,8], **Bocharov V.** (victor.bocharov@gmail.com)[3,5], **Alexeeva S.** (sv.bichineva@gmail.com)[3], **Droganova K.** (kira.droganova@gmail.com)[4,9], **Fenogenova A.** (alenka_s_ph@mail.ru)[4,7], **Granovsky D.** (dima.granovsky@gmail.com)[3]

[1]Lomonosov Moscow State University, [2]MIPT, [3]OpenCorpora.org, [4]National Research University Higher School of Economics, [5]Yandex, [6]GICR, [7]RDI KVANT, [8]Vinogradov Institute of the Russian Language RAS, [9]Charles University

MorphoRuEval-2017 is an evaluation campaign designed to stimulate the development of the automatic morphological processing technologies for Russian, both for normative texts (news, fiction, nonfiction) and those of less formal nature (blogs and other social media). This article compares the methods participants used to solve the task of morphological analysis. It also discusses the problem of unification of various existing training collections for Russian language.

LEVENSHTEIN DISTANCE AND WORD ADAPTATION SURPRISAL AS METHODS OF MEASURING MUTUAL INTELLIGIBILITY IN READING COMPREHENSION OF SLAVIC LANGUAGES

Stenger I. (ira.stenger@mx.uni-saarland.de), **Avgustinova T.** (avgustinova@coli.uni-saarland.de), **Marti R.** (rwmslav@mx.uni-saarland.de), Saarland University, Saarbrücken, Germany

In this article we validate two measuring methods: Levenshtein distance and word adaptation surprisal as potential predictors of success in reading intercomprehension. We investigate to what extent orthographic distances between Russian and other East Slavic (Ukrainian, Belarusian) and South Slavic (Bulgarian, Macedonian, Serbian) languages found by means of the Levenshtein algorithm and word adaptation surprisal correlate with comprehension of unknown Slavic languages on the basis of data obtained from Russian native speakers in online free translation task experiments. We try to find an answer to the following question: Can measuring methods such as Levenshtein distance and word adaptation surprisal be considered as a good approximation of orthographic intelligibility of unknown Slavic languages using the Cyrillic script?

COREFERENCE RESOLUTION IN RUSSIAN: STATE-OF-THE-ART APPROACHES APPLICATION AND EVOLVEMENT

Sysoev A. A. (sysoev@ispras.ru), **Andrianov I. A.** (ivan.andrianov@ispras.ru), **Khadzhiiskaia A. Y.** (sanya@ispras.ru), Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia

Coreference resolution aims at grouping textual references denoting same real world entities into clusters. Many state-of-the-art results have already been received for coreference resolution in European languages, but for Russian this area is still quite novel and underexplored. With this paper we try to fill this gap. Our article reviews existing approaches and presents their adaptation for Russian language. We carry out sufficient number of experiments to estimate efficiency of various machine learning methods and features, utilized under the hood of the algorithms. Additionally we propose a novel feature to be used for head detection subtask, which is based on word embeddings clustering. As a result, we managed to establish baseline implementation for Russian language coreference resolution problem. The key features of the developed approach are simplicity and extensibility. Presence of such a baseline opens many research directions for improving quality of the algorithms; some potential improvements are already pointed out in this paper. We expect further works in this area to significantly increase current level of state-of-the-art results for Russian coreference resolution, making it practically applicable in the near future.

NEURAL NETWORK BASED END-TO-END LEARNING HIERARCHY-AWARE SEMANTIC PARSER FOR RUSSIAN LANGUAGE

Tarasov D. S. (dtarasov@meanotek.io), **Lukina N. M.**, **Izotova E. D.**, Meanotek AI Research, Kazan, Russia

We present neural network semantic parser for Russian language, that utilizes new copying mechanism, intermediate layers supervision and explicit handling of hierarchical nature of the output by means of having RNN blocks operating on different timeframes. Due to the lack of standard Russian dataset for validating semantic parsers, we develop our own small dataset in the domain of logistics and task management and demonstrate that our model can obtains good results on this dataset, despite it very limited size.

COREFERENCE RESOLUTION FOR RUSSIAN: THE IMPACT OF SEMANTIC FEATURES

Toldova S. (toldova@yandex.ru), National Research University Higher School of Economics, Moscow, Russia; **Ionov M.** (max.ionov@gmail.com), Lomonosov Moscow State University, Moscow, Russia

This paper presents the results of our experiments on building a general coreference resolution system for Russian. The main aim of those experiments was to set a baseline for this task for Russian using the standard set of features developed and tested for coreference resolution systems created for other languages. We propose several baseline systems, both rule-based and ML-based. We show that adding some semantic information is crucial for the task and even the small amount of data can improve the overall result. We show that different types of semantic resources affect the performance differently and sometimes more does not imply better.

A SYNTAX-BASED DISTRIBUTIONAL MODEL FOR DISCRIMINATING BETWEEN SEMANTIC SIMILARITY AND ASSOCIATION

Trofimov I. V. (itrofimov@gmail.com), **Suleymanova E. A.** (yes2helen@gmail.com),
Program Systems Institute of RAS, Pereslavl-Zalessky, Russia

In recent years, distributional semantics has shown a trend towards a deeper understanding of what semantic relatedness is and what it is composed of. This is attested, in particular, by the emergence of new gold standards like SimLex999, WS-Sim and WS-Rel. Evidence from cognitive psychology suggests that humans distinguish between two basic types of semantic relations: category-based similarity and thematic association. The paper presents a distributional model capable of differentiating between these relations, and a dataset consisting of 500 similar and 500 associated pairs of nouns that can be used for evaluation of such models.

EXPANDING HIERARCHICAL CONTEXTS FOR CONSTRUCTING A SEMANTIC WORD NETWORK

Ustalov D. A. (dau@imm.uran.ru), Krasovskii Institute of Mathematics and Mechanics;
Ural Federal University, Yekaterinburg, Russia

A semantic word network is a network that represents the semantic relations between individual words or their lexical senses. This paper proposes Watlink, an unsupervised method for inducing a semantic word network (SWN) by constructing and expanding the hierarchical contexts using both the available dictionary resources and distributional semantics' methods for *is-a* relations. It has three steps: context construction, context expansion, and context disambiguation. The proposed method has been evaluated on two different datasets for the Russian language. The former is a well-known lexical ontology built by the group of expert lexicographers. The latter, LRWC ("Lexical Relations from the Wisdom of the Crowd"), is a new resource created using crowdsourcing that contains both positive and negative human judgements for subsumptions. The proposed method outperformed the other relation extraction methods on both datasets according to recall and F_1-score. Both the implementation of the Watlink method and the LRWC dataset are publicly available under libré licenses.

MULTI-LEVEL STUDENT ESSAY FEEDBACK IN A LEARNER CORPUS

Vinogradova O. I. (olgavinogr@gmail.com)[1], **Lyashevskaya O. N.** (olesar@yandex.ru)[1,2],
Panteleeva I. M. (irapanteleeva@rambler.ru)[1]
[1]National Research University Higher School of Economics, [2]Vinogradov Institute of the Russian Language RAS, Moscow, Russia

The paper presents the results of using computer tools and of designing an inspection program for the purposes of the automated and semi-automated syntactic, lexical, and grammar error analysis of student essays in a learner corpus. The texts in the corpus were written in English by Russian learners of English. In our experiment we compare the parameters of the essays graded by professional examiners as the best and those graded the lowest in the pool of about 2000 essays. At the first stage in the experiment we applied a syntactic tool for parsing the sentences and collected data regarding mean sentence depth and the average number of relative, other adnominal, and adverbial clauses, then analyzed the results of lexical observations in those texts (such as average word length, number of academic words, number of linking words and some others), and finally collected the statistics related to the errors pointed out in manual expert annotation. The parameters that had very different values for the "good" and for the "bad" essays are regarded by the authors as worthy parts of the feedback a student can get for the text uploaded into the learner corpus.

AUTOMATIC COLLOCATION EXTRACTION: ASSOCIATION MEASURES EVALUATION AND INTEGRATION

Zakharov V. P. (v.zakharov@spbu.ru), Saint-Petersburg State University, Saint-Petersburg, Russia

The paper deals with collocation extraction from corpus data. A collocation is meant as a special type of a set phrase. Many modern authors and most of corpus linguists understand collocations as statistically determined set phrases. The above approach is the basic point of this paper which is aimed at evaluation of various statistical methods of automatic collocation extraction. There are several ways to calculate the degree of coherence of parts of a collocation. A whole number of formulae have been created to integrate different factors that determine the association between the collocation components. Usually, such formulae are called association measures.

The experiments are described which objective was to study the method of collocation extraction based on the statistical association measures. We extracted collocations for the word *вода* (water) and some others by means of the tool Collocations of the NoSketch Engine system using 7 association measures. It is important to stress that the experiments were conducted using representative corpora, with large amount of the resulting collocations being under study. The data on the measure precision allows to establish to some degree that in cases when collocation extraction is not used for some special purposes such measures as *MI.l-og_f, log-Dice*, and *minimum sensitivity* should be used. No measure is ideal, which is why various options of their integration are desirable and useful. And we propose a number of parameters that allow to rank collocates in an integrated list, namely, an average rank, a normalised rank and an optimised rank.

PARAPHRASED PLAGIARISM DETECTION USING SENTENCE SIMILARITY

Zubarev D. V. (dvzubarev@yandex.ru), **Sochenkov I. V.** (isochenkov@sci.pfu.edu.ru)
RUDN University, Moscow, Russia; Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

The paper describes an approach to plagiarism detection within PlagEvalRus-2017 competition. Our system leverages deep parsing techniques to be able to detect moderately disguised plagiarism. We participated in the two tracks of the competition: source retrieval (sources detection) and text alignment (paraphrased plagiarism detection). There are various cases of plagiarism presented in datasets of both tracks. They vary by the level of disguise that was used while reusing text. The results show that our method performed quite well for detecting moderately disguised forms of plagiarism.

Авторский указатель

Author Index

Научное издание

Компьютерная лингвистика и интеллектуальные технологии

По материалам ежегодной
международной конференции «Диалог»

Выпуск 16 (23). 2017

Том 1

Ответственный за выпуск **А. А. Белкина**
Вёрстка **К. А. Климентовский**
Дизайн обложки **А. А. Светличная**

Подписано в печать 12.05.2017
Формат 152 × 235
Бумага офсетная
Тираж 350 экз. Заказ № 79

Издательский центр «Российский
государственный гуманитарный университет»
125993, Москва, Миусская пл., д. 6
Тел.: +7 499 973 42 06

Отпечатано с готового оригинал-макета в типографии
ООО «Издательско-полиграфический центр Маска»
117246, Москва, Научный пр-д, д. 20, стр. 9