Компьютерная лингвистика и интеллектуальные технологии

По материалам ежегодной Международной конференции «Диалог» (2015)

Выпуск 14

В двух томах

Том 2. Доклады специальных секций

Computational Linguistics and Intellectual Technologies

Papers from the Annual International Conference “Dialogue” (2015)

Issue 14

Volume 2 of 2. Papers from special sessions

УДК 80/81; 004
ББК 81.1
К63

Программный комитет конференции выражает
искреннюю благодарность Российскому фонду
фундаментальных исследований за финансовую поддержку,
грант № 15-07-20554 Г

Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Москва, 27–30 мая 2015 г.). Вып. 14 (21): В 2 т. Т. 2: Доклады специальных секций. — М.: Изд-во РГГУ, 2015.

Сборник включает 69 докладов международной конференции по компьютерной лингвистике и интеллектуальным технологиям «Диалог 2015», 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

14-й выпуск ежегодника «Компьютерная лингвистика и интеллектуальные технологии» содержит избранные материалы 21-й Международной конференции «Диалог». На основании мнений наших рецензентов для публикации в ежегоднике Редсоветом было отобрано 69 докладов из числа примерно ста работ, которые были рекомендованы по результатам рецензирования для представления на конференции в этом году.

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

- Формальные модели языка и их применение в компьютерной лингвистике
- Модели и методы семантического анализа текста
- Лингвистические онтологии и извлечение знаний
- Теоретическая и компьютерная лексикография
- Методики тестирования технологий и верификации результатов лингвистических исследований (Dialogue Evaluation)
- Компьютерные лингвистические ресурсы и их связывание (Linked Data)
- Корпусная лингвистика: создание, разметка, методики применения и оценка корпусов
- Анализ Social Media
- Машинный перевод текста и речи
- Лингвистический анализ речи
- Модели общения. Коммуникация, диалог и речевой акт
- Мультимодальная лингвистика
- Компьютерный анализ документов: классификация, поиск, тематический анализ, оценка тональности и т. д.

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

В этом году продолжается и традиция проведения в рамках направления Dialogue Evaluation тестирований технологий решения отдельных задач компьютерного анализа русского языка. Значимость таких мероприятий трудно переоценить, поскольку их результаты создают основу для сравнительной оценки эффективности результатов в соответствующих областях исследований.

В этом году было проведено два таких тестирования: сравнивались различные подходы к анализу т.н. аспектного сентимента и оценке семантической близости слов.

Sentiment Analysis является важным самостоятельным прикладным направлением компьютерной лингвистики, особенно в той постановке, которая была предложена участникам: не только определение общей тональности документа, но и выделение и оценка в нем отдельных аспектов выражения мнения.

Тестирование методов определения семантической близости слов является важным для понимания сложной картины в современной вычислительной семантике, где конкурируют и взаимодействуют традиционные словарные и новые дистрибуционные методы исследования лексических значений.

Наиболее значимые работы, представленные участниками этих тестирований, выделены в отдельный второй том ежегодника. Там же опубликованы и итоговые статьи организаторов.

Программный комитет конференции выражает особую признательность Наталье Лукашевич и Александру Панченко за особую роль в организации и проведении этих тестирований.

Среди особых направлений «Диалога» в этом году — исследования в области русского мультимодального дискурса. Интерес к языковой коммуникации как целому всегда был характерным для нашей конференции, выросшей, напомним, из семинара, носившего название «Модели общения». Доклады мультимодального направления составляют важную часть этого сборника.

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 языке).

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.

Программный комитет «Диалога»
Редколлегия ежегодника «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
- Компания Yandex
- Филологический факультет МГУ

Конференция проводится при поддержке Российской ассоциации искусственного интеллекта.

Международный программный комитет

Байтин Алексей Владимирович	Компания Yandex
Богуславский Игорь Михайлович	Политехнический университет Мадрида
Буате Кристиан	Гренобльский университет
Гельбух Александр Феликсович	Национальный политехнический институт, Мехико
Иомдин Леонид Лейбович	Институт проблем передачи информации РАН
Кобозева Ирина Михайловна	Филологический факультет МГУ
Козеренко Елена Борисовна	Институт проблем информатики РАН
Корбетт Гревил	University of Surrey, UK
Кронгауз Максим Анисимович	Институт Лингвистики РГГУ
Лукашевич Наталья Валентиновна	НИВЦ МГУ
Маккарти Диана	Lexical Computing Ltd., UK
Мельчук Игорь Александрович	Монреальский университет
Ниврe Йоаким	Уппсальский университет
Ниренбург Сергей	Университет Нью-Мексико
Осипов Геннадий Семёнович	Институт программных систем РАН
Попов Эдуард Викторович	РосНИИ информационной техники и САПР
Раскин Виктор	Purdue University, USA
Селегей Владимир Павлович	Компания ABBYY
Хови Эдуард	University of Southern California
Шаров Сергей Александрович	University of Leeds, UK

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	Kontur Labs; Уральский федеральный университет
Добров Борис Викторович	НИВЦ МГУ
Иомдин Леонид Лейбович	Институт проблем передачи информации РАН
Кобозева Ирина Михайловна	Филологический факультет МГУ
Козеренко Елена Борисовна	Институт проблем информатики РАН
Лауфер Наталия Исаевна	ООО «проФан Продакшн»
Ляшевская Ольга Николаевна	Universitet i Tromsø, Norway
Сердюков Павел Викторович	Компания Yandex
Соколова Елена Григорьевна	РосНИИ искусственного интеллекта
Толдова Светлана Юрьевна	Филологический факультет МГУ
Шаров Сергей Александрович	University of Leeds, UK

Секретариат

Белкина Александра Андреевна, *секретарь оргкомитета*	Компания ABBYY
Атясова Анастасия Леонидовна, *координатор*	Компания ABBYY

Рецензенты

Августинова Таня
Азарова Ирина Владимировна
Апресян Валентина Юрьевна
Байтин Алексей Владимирович
Баранов Анатолий Николаевич
Беликов Владимир Иванович
Богданов Алексей Владимирович
Богданова-Бегларян Наталья Викторовна
Богуславский Игорь Михайлович
Бонч-Осмоловская Анастасия Александровна
Браславский Павел Исаакович
Васильев Виталий Геннадьевич
Гельбух Александр Феликсович
Гершман Анатолий
Гращенков Павел Валерьевич
Гриненко Михаил Михайлович
Гришина Елена Александровна
Губин Максим Вадимович
Даниэль Михаил Александрович
Добров Борис Викторович
Добровольский Дмитрий Олегович
Добрынин Владимир Юрьевич
Зализняк Анна Андреевна
Захаров Виктор Павлович
Захаров Леонид Михайлович
Иванов Владимир Владимирович
Иомдин Борис Леонидович
Иомдин Леонид Лейбович
Кибрик Андрей Александрович
Кобозева Ирина Михайловна
Козеренко Елена Борисовна
Коротаев Николай Алексеевич
Котельников Евгений Вячеславович
Котов Артемий Александрович
Крейдлин Григорий Ефимович
Кронгауз Максим Анисимович
Леонтьев Алексей Петрович
Лобанов Борис Мефодьевич
Лукашевич Наталья Валентиновна
Ляшевская Ольга Николаевна
Маккарти Диана
Минлос Филипп Робертович
Недолужко Анна Юрьевна
Новицкий Валерий Игоревич
Пазельская Анна Германовна
Панченко Александр Иванович
Паперно Денис Аронович
Пиперски Александр Чедович
Подлесская Вера Исааковна
Савельев Василий Евгеньевич
Селегей Владимир Павлович
Смирнов Иван Валентинович
Сокирко Алексей Викторович
Соколова Елена Григорьевна
Сорокин Алексей Андреевич
Старостин Анатолий Сергеевич
Тихомиров Илья Александрович
Толдова Светлана Юрьевна
Турдаков Денис Юрьевич
Урысон Елена Владимировна
Федорова Ольга Викторовна
Хови Эдуард
Хорошевский Владимир Федорович
Циммерлинг Антон Владимирович
Шаров Сергей Александрович
Янко Татьяна Евгеньевна

Contents*

Раздел II.
Анализ тональности

* The reports of each section are ordered by the surname of the first author in compliance with the English alphabet.

Раздел III.
Анализ семантической близости

Раздел II.

Анализ тональности

SENTIRUEVAL: ТЕСТИРОВАНИЕ СИСТЕМ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТОВ НА РУССКОМ ЯЗЫКЕ ПО ОТНОШЕНИЮ К ЗАДАННОМУ ОБЪЕКТУ

Лукашевич Н. В. (louk_nat@mail.ru)[1],
Блинов П. Д. (blinoff.pavel@gmail.com)[2],
Котельников Е. В. (kotelnikov.ev@gmail.com)[2],
Рубцова Ю. В. (yu.rubtsova@gmail.com)[3],
Иванов В. В. (nomemm@gmail.com)[4],
Тутубалина Е. В. (tlenusik@gmail.com)[4]

[1]МГУ им. М. В. Ломоносова, Москва, Россия;
[2]Вятский государственный гуманитарный университет, Киров, Россия;
[3]Институт систем информатики им. А. П. Ершова СО РАН, Новосибирск, Россия;
[4]Казанский федеральный университет, Казань, Россия

Статья описывает данные, правила и результаты SentiRuEval — тестирования систем автоматического анализа тональности русскоязычных текстов по отношению к заданному объекту или его свойствам. Участникам были предложены два задания. Первое задание было аспектно-ориентированный анализ отзывов о ресторанах и автомобилях; основная цель этого задания была найти слова и выражения, обозначающие важные характеристики сущности (аспектные термины), и классифицировать их по тональности и обобщенным категориям. Второе задание заключалось в анализе влияния твитов на репутацию заданных компаний. Такие твиты могут либо выражать мнение пользователя о компании, ее продукции или услугах, или содержать негативные или позитивные факты, которые стали известны об этой компании.

Ключевые слова: анализ тональности текстов, оценка качества, разметка коллекций, оценочные слова

SENTIRUEVAL: TESTING OBJECT-ORIENTED SENTIMENT ANALYSIS SYSTEMS IN RUSSIAN

Loukachevitch N. V. (louk_nat@mail.ru)[1],
Blinov P. D. (blinoff.pavel@gmail.com)[2],
Kotelnikov E. V. (kotelnikov.ev@gmail.com)[2],
Rubtsova Y. V. (yu.rubtsova@gmail.com)[3],
Ivanov V. V. (nomemm@gmail.com)[4],
Tutubalina E. V. (tlenusik@gmail.com)[4]

[1]Lomonosov Moscow State University, Moscow, Russia;
[2]Vyatka State Humanities University, Kirov, Russia;
[3]A. P. Ershov Institute of Informatics Systems, Novosibirsk, Russia;
[4]Kazan Federal University, Kazan, Russia

The paper describes the data, rules and results of SentiRuEval, evaluation of Russian object-oriented sentiment analysis systems. Two tasks were proposed to participants. The first task was aspect-oriented analysis of reviews about restaurants and automobiles, that is the primary goal was to find word and expressions indicating important characteristics of an entity (aspect terms) and then classify them into polarity classes and aspect categories. The second task was the reputation-oriented analysis of tweets concerning banks and telecommunications companies. The goal of this analysis was to classify tweets in dependence of their influence on the reputation of the mentioned company. Such tweets could express the user's opinion or a positive or negative fact about the organization.

Keywords: sentiment analysis, users review, collection labeling, aspect words, evaluation

1. Introduction

During last years the task of automatic sentiment analysis of natural language texts, that automatic extraction of opinions expressed in texts, attracts a lot of attention of researchers and practitioners. This is due to the fact that this task has a lot of useful applications. So the analysis and representation of users' opinions about products and services are of interest to their producers and competitors as well as to new users. Social opinion processing is important for authorities for better government.

The initial approaches to automatic sentiment analysis tried to determine the overall sentiment of the whole texts or sentences (Pang et al., 2002). This level of analysis presupposes that each document expresses opinions on a single entity (for example, a single product). Later, the task of object-oriented sentiment analysis appeared, when the system should reveal sentiment towards a specific entity mentioned in the text (Amigo et al., 2012; Jiang et al., 2011).

Finally, an author of a text can have different opinions relative to specific properties (or aspects) of an entity. To reveal these opinions, so called aspect-based sentiment analysis should be fulfilled (Liu, 2012; Bagheri et al., 2013; Glavaš et al., 2013; Popescu, Etzioni, 2005; Zhang, Liu, 2014). Aspects are expressed in texts with aspect terms and usually can be classified into categories. For example, "Service" aspect category in restaurant reviews can be expressed such terms as *staff*, *waiter*, *waitress*, *server*.

Automatic sentiment analysis is a complex problem of natural language processing. Several evaluation initiatives were devoted to study the best methods in sentiment analysis and related applications. These initiatives include Blog Track within TREC conference (Macdonald et al., 2010), TAC Opinion QA Tasks (Dang, Owczarzak, 2008), opinion tracks at NTCIR conferences (Seki et al., 2008), reputation management tracks at CLEF conference (Amigo et al., 2012), Twitter and review sentiment analysis tasks within SemEval initiative (Nakov et al., 2013; Rosenthal et al., 2014), etc.

In this paper we present results of SentiRuEval evaluation focusing on entity-oriented sentiment analysis of Twitter and aspect-oriented analysis of users' reviews in Russian. This evaluation is the second Russian sentiment analysis evaluation event in Russian after ROMIP sentiment analysis tracks in 2011–2013. This year in SentiRuEval we had two types of tasks. The first task is aspect-oriented sentiment analysis of users' reviews. The data included reviews about restaurants and automobiles. The second task was object-oriented sentiment analysis of Russian tweets concerning two varieties of organizations: banks and telecommunications companies.

The structure of this paper is as follows. In Section 2 we consider related evaluation initiatives in sentiment analysis. Section 3 describes tasks, data and principles of labeling in aspect-based review analysis. Section 4 describes the data and the task in the entity-oriented sentiment analysis of Twitter. Section 5 discusses results obtained by participants.

2. Related work

Several evaluation initiatives were devoted to sentiment analysis tasks similar to current SentiRuEval evaluation.

Last years in the framework of SemEval conference two types of sentiment analysis evaluations have been organized: sentiment analysis in Twitter and aspect-based sentiment analysis of reviews. In the Twitter task one of the subtasks was a message-level task, that is participating systems should classify if the message has positive, negative, or neutral sentiment (Nakov et al., 2013; Rosenthal et al., 2014). The task is directed to reveal, namely, the author opinion in contrast to neutral or objective information.

In the framework of CLEF initiative (http://www.clef-initiative.eu/) in 2012–2014 RebLab evaluations devoted to monitoring of reputation-oriented tweets were organized. The tasks included the definition of the polarity for reputation classification. The goal was to decide if the tweet content has positive or negative implications for the company's reputation. The organizers stress that the polarity for reputation is substantially different from standard sentiment analysis that should differentiate subjective

from objective information. When analyzing polarity for reputation, both facts and opinions have to be considered to determine what implications a piece of information might have on the reputation of a given entity (Amigo et al., 2012; Amigo et al., 2013).

Evaluation of aspect-based review analysis at SemEval was organized in 2014 for the first time (Pontiki et al., 2014). The dataset included isolated, out of context sentences (not full reviews) in two domains: restaurants and laptops. 3K sentences were prepared for training in each domain. Set of aspect categories for restaurants included: *food, service, price, ambience, anecdotes/miscellaneous*.

In 2015 SemEval evaluations the aspect-based sentiment analysis of reviews (http://alt.qcri.org/semeval2015/task12/) is focused on entire reviews. Aspect categories of terms became more complicated and now consist of Entity-Attribute pairs (E#A). The E#A inventories for the restaurants domain contains 6 Entity types (RESTAURANT, FOOD, DRINKS, SERVICE, AMBIENCE, LOCATION) and 5 Attribute labels (GENERAL, PRICES, QUALITY, STYLE_OPTIONS, MISCELLANEOUS). The Laptops domain contains 22 Entity types and 9 Attribute labels.

In 2011–2013 two evaluation events of Russian sentiment analysis systems were organized. The first evaluation was devoted to extraction of overall sentiment of users' reviews in three domains: movies, books and digital cameras. For training, reviews from recommendation services were granted to participants. The evaluation was fulfilled on blog posts extracted with the help of the Yandex blog service (Chetviorkin et al., 2012). The second evaluation offered two new tasks for participants, namely: extraction of the overall sentiment of quotation (direct or indirect speech) from news articles and sentiment-oriented information retrieval in blogs when for a query (from the abovementioned domains) user opinions in blog posts should be found (Chetviorkin, Loukachevitch, 2013).

3. Ways to express opinions about aspects

Aspect terms also can be subdivided into several categories. They can be classified into three subtypes: **explicit aspects**, **implicit aspects** and **sentiment facts**.

Explicit aspects denote some part or characteristics of a described object such as *staff, pasta, music* in restaurant reviews. Explicit aspects are usually nouns or noun groups, but in some aspect categories we can meet explicit aspects expressed as verbs. For example, in restaurants the important characteristics of the service quality is time of order waiting, so this characteristic can be mentioned with verb *wait* (*ждать*): *ждали больше часа—waited for more than an hour.*

Implicit aspects are single words or single words with sentiment operators that contain within themselves as specific sentiments as the clear indication to the aspect category. In restaurant reviews the frequent implicit aspects are such words as *tasty* (*positive+food*), *comfortable* (*positive+interior), not comfortable* (*negative+interior*). The importance of these words for automatic systems consists in that fact that implicit aspects allow a sentiment system to reveal user's opinion about entity characteristics even if an explicit aspect term is unknown, written with an error or referred in a complicated way.

Sentiment facts do not mention the user sentiment directly, formally they inform us only about a real fact, however, this fact conveys us a user's sentiment as well as the aspect category it related to. For example, sentiment fact *отвечала на все вопросы* (*answered all questions*) means positive characterization of the restaurant service; this expression is enough frequent in restaurant reviews.

In the SentiRuEval labeling we annotated these three subtypes of aspect terms and our tasks for participants were not only to extract explicit aspect terms but also to extract all aspect terms (see Section 4).

An opinion about aspects can be expressed in several ways.

The **direct way of conveying the opinion** is through using opinion words such as *good*, *bad*, *excellent*, *awful*, *like*, *hate*, etc.

Opinions can be formulated as **comparisons** with other entities, previous cases or opinions of other people (Liu, 2012; Jindal, Liu, 2006). The problem of automatic analysis in these cases arise because used positive or negative words can be not relevant to the current review. In addition, comparison can be delivered in various ways not only using comparative constructions. For example, in the following extract from a restaurant review the comparison is marked with word *another,* and positive words *enjoyed* and *wonderful* characterize a restaurant distinct from the restaurant under review:

> *We decided not to have dessert and coffee there, but instead went to another restaurant where we* ***enjoyed*** *a* ***wonderful*** *end to our evening.*

We can formulate our opinion as **recommendation** (the constructive or suggestive opinion—see (Arora, Srinivasa, 2014)) or description of a **desirable situation** or characteristics of an entity, so called *irrealis factors* (Taboada et al., 2011; Kusnetsova et al., 2013). In these cases mentioned positive words can conceal the negative opinion.

At last, the opinion can be expressed with means of **irony or sarcasm** (Barbieri, Saggion, 2014; Riloff et al., 2013). In such cases the opinion can look like positive or at least medium one, but in fact it is strongly negative as in the following example: "***Excellent*** *translation, I don't understand anything*".

In the SentiRuEval labeling we marked these subtypes of opinions for further research (see Section 4).

4. Labeling and tasks of aspect-based analysis of reviews at SentiRuEval

For evaluation of aspect-oriented sentiment analysis systems we chose two domains: restaurant reviews and automobile reviews. In restaurant reviews aspect categories include: FOOD, SERVICE, INTERIOR (including atmosphere), PRICE, GENERAL. For automobiles aspect categories are: DRIVEABILITY, RELIABILITY, SAFETY, APPEARANCE, COMFORT, COSTS, GENERAL.

The length of reviews can vary drastically from one brief sentence to a long narrative. There can be also shifts to one or the other particular aspect. As an experiment, for labeling in the restaurant domain we tried to extract the most typical reviews

from our collection. To achieve it, the following procedure was performed. We represented each review as a bag-of-word vector and calculated the global collection's vector by averaging all the individual vectors. Then we imposed restrictions on min and max review length and chose most similar reviews according to the cosine similarity between global vector and single review vectors. As a result, most typical review representatives were selected for the labeling.

The labeling of training and test data was conducted with BRAT annotating tool (Stenetorp et al., 2012). Annotators had access to review collections through web interface. To unify and agree the annotation procedure, an assessor manual was prepared[1]. It is based on the SemEval-2014 (Pontiki et al., 2014) annotation guidelines.

The annotation task was to mark up two main types of tokens: aspect terms within a review and aspect categories attached to whole reviews. The aspect categories were labeled with the overall score of sentiment expressed in the text: positive, negative, both or absent.

According to the above-described categorization of opinions and aspect terms, the annotation of aspect terms within a text included several dimensions:

1. At first annotators should indicate explicit aspects, implicit aspects or sentiment facts in review texts and assign them their relevant type (explicit, implicit or fact).
2. All aspects terms should be assigned to aspect categories of the target entity.
3. Annotators marked the polarity of the aspect term: positive, negative, neutral, or both.
4. Annotators marked the relevance of the term to the review:
    a. *Rel—relevant* (to the current review),
    b. *Cmpr—comparison*, that is the term concerns another entity,
    c. *Prev—previous*, that is the term is related to previous opinions,
    d. *Irr—irrealis*, that is the term is the part of a recommendation or description of a desirable situation,
    e. *Irn—irony*.

So, for example, the annotation of word *девушка* (*girl*) in context *милая девушка* (*nice girl*) in a restaurant review includes sentiment orientation—*positive*, aspect category—*service*, aspect mark—*relevant,* aspect type—*explicit*.

Such detailed annotation process is very labor consuming. Therefore, each review was labeled only by a single assessor. However, to check the quality of aspect labeling two procedures were fulfilled after the labeling was finished. First, all labeled aspect terms were extracted from the markup according to their types and categories and were looked through; so some accidental mistakes were found and corrected. Second, we compared the aspect sentiment assigned to the review as a whole and sentiments of specific terms within this review. In cases of the differences between these two types of labeling the markup of the review was additionally verified.

During the annotation procedure, no balancing according to sentiment or aspect terms was performed; we tried to keep natural distributions specific for reviews in a given domain. Some statistics about relevant terms (Rel) are shown in Table 1.

---

[1] The manual is available at http://goo.gl/Wqsqit.

Table 1. Corpus statistics

		Restaurants		Automobiles	
		Train	**Test**	**Train**	**Test**
Number of reviews		201	203	217	201
Number of terms which are	explicit	2,822	3,506	3,152	3,109
	implicit	636	657	638	576
	fact	523	656	668	685
Number of terms which are	positive	2,530	3,424	2,330	2,499
	negative	684	865	1,337	1,300
	neutral	714	445	691	456
	both	53	85	100	115

The labeled data allowed us to offer the following tasks to the participants:

- **Task A**: automatic extraction of explicit aspects,
- **Task B**: automatic extraction of all aspects including sentiment facts,
- **Task C**: extraction of sentiments towards explicit aspects,
- **Task D**: automatic categorization of explicit aspects into aspect categories,
- **Task E**: sentiment analysis of the whole review on aspect categories.

To evaluate automatic systems the following quality measures were utilized.

For task A and B we applied macro F1-measure in two variants: exact matching and partial matching. Macro F1-measure means in this case calculating F1-measure for every review and averaging the obtained values.

To measure partial matching for every gold standard aspect term t we calculate precision and recall in the following way:

$$\text{Precision}_t = \frac{|t \cap t_s|}{|t_s|},$$

$$\text{Recall}_t = \frac{|t \cap t_s|}{|t|},$$

where t_s is an extracted aspect term that intersects with term t, $t \cap t_s$ is the intersection between terms t and t_s, $|t|$ is the length of the term in tokens. So F1-measure is calculated for every term and then we average the values for all gold standard terms.

For sentiment classification of aspect terms (task C) both variants of F1-measure (macro- and micro-) were utilized. Calculation of macro F1-measure is based on separate calculation of precision, recall, and F-measure for every category under consideration, then the obtained values are averaged. This allows us to evaluate the quality of categorization equally for every category. Micro F1-measure is calculated on the global confusion matrix, this measure greatly depends on the disbalance in the class distribution.

For aspect categorization of terms (task D) and the sentiment analysis of whole reviews (task E) macro F1-measure was used.

Table 2. Results in aspect-oriented review analysis (Restaurant domain)

Task	Measure	Baseline	Participants' results	Participant identifier
A	Exact matching, Macro F	0.608	**0.632**	**2**
			0.627	1
A	Partial matching, Macro F	0.665	**0.728**	**4**
			0.719	1
B	Exact matching, Macro F	0.587	**0.600**	**1**
			0.596	2
B	Partial matching, Macro F	0.619	**0.668**	**1**
			0.645	1
C	Macro F	0.267	**0.554**	**4**
			0.269	3
C	Micro F	0.710	**0.824**	**4**
			0.670	3
D	Macro F	0.800	**0.865**	**8**
			0.810	4
E	Macro F	0.272	**0.458**	**4**
			0.372	10

For all tasks we prepared baseline runs. The baseline system for tasks A and B extracts the list of labeled terms from the training collection, lemmatizes them and apply them to the lemmatized representation of the test collection. If more than one term matches the same word sequence, then a longer term is preferred.

The task C and D baseline systems attribute an aspect term to its most frequent category in the training collection. If a term is absent in the training collection then the most frequent aspect category is applied. The task E baseline is the most frequent sentiment category for the given aspect category (positive in all cases).

Altogether 12 participants with 21 runs were participated in the review sentiment analysis tasks. Due space limitations here we represent only two best results in each task and only primary F-measure, the full results are available at http://goo.gl/Wqsqit. Table 2 presents the participants' results for restaurant reviews, Table 3 contains the results for automobile reviews. Automobile reviews obtained much less attention from participants.

From the Tables 2, 3 it can be seen that the baselines for extracting aspect terms (tasks A and B) are quite high, which means the considerable agreement between annotation of training and testing collections. The best methods in these tasks were based on distributional approaches augmented with a set of rules (participant 4) and recurrent neural nets (participant 1). For the exact aspect matching, the best results were achieved by sequence labeling with SVM on the rich set of morphological, syntactic and semantic features (participant 2).

Table 3. Results in aspect-oriented review analysis (Automobile domain)

Task	Measure	Baseline	Participants' results	Participant identifier
A	Exact matching, Macro F	0.594	**0.676**	**2**
			0.651	1
A	Partial matching, Macro F	0.697	**0.748**	**1**
			0.730	2
B	Exact matching, Macro F	0.589	**0.636**	**2**
			0.630	1
B	Partial matching, Macro F	0.674	**0.714**	**1**
			0.704	1
C	Macro F	0.264	**0.568**	**4**
			0.342	1
C	Micro F	0.619	**0.742**	**4**
			0.647	1
D	Macro F	0.564	**0.652**	**8**
			0.607	4
E	Macro F	0.237	0.439	4

The best result in the analysis of sentiment towards aspect terms (task C) was obtained with Gradient Boosting Classifier (participant 4). The features were based on the skip-gram model exploiting word contexts for learning better vector representations and pointwise mutual information. In the task of categorization of explicit aspect terms (task D) the best results were obtained by SVM with features based on pointwise mutual information (participant 8). The second-place result is obtained by the method relying on the term similarity in the space of distributed representations of words (participant 4). For task E the best results were achieved by integration of the results obtained in tasks A, C and D (participant 4).

5. Object-oriented sentiment analysis of tweets

The goal of Twitter sentiment analysis at SentiRuEval was to find sentiment-oriented opinions or positive and negative facts about two types of organizations: banks and telecom companies. This task is quite similar to the reputation polarity task at RepLab evaluation (Amigo et al., 2013).

The training and test tweet collections were provided with fields corresponding all possible organizations for that tweets were extracted. A concrete organization mentioned in a given tweet was indicated with "0" label, denoting "neutral" as a default value. Annotators and participating systems should to leave this value unchanged if the tweet was considered as neutral or replace the value with "1" (positive) or "−1" (negative). The annotators also could label tweets with "−−", which means =meaningless=, or with "+−", which means positive and negative sentiments in the same tweet. Both latter cases were excluded from evaluation.

For training and testing collections assessors labeled 5,000 tweets in each domains (20000 tweets were labeled altogether). It is important to stress, that the training and testing collections were issued during different time intervals. The tweets of the training collection were written in 2014, the tweets of the testing collection were published in 2013.

Table 4. Results of the voting procedure in labeling of the tweet testing collection

Domain	The number of tweets with the same labels from at least 2 assessors	Full coincidence of labeling	The final number of tweets in the testing collection
Banks	4,915 (98.30%)	3,816 (76.36%)	4,549
Telecom companies	4,503 (90.06%)	2,233 (44.66%)	3,845

Analyzing the markup of the training collection we found that the estimation of some tweets can arise considerable discussion on their sentiment. To lessen the subjectivity of labeling and also accidental mistakes the testing collection was labeled by three assessors, and the voting scheme was applied to obtain the results of manual labeling. Finally, from the collection irrelevant tweets were removed. Results of the preparing the collection are presented in Table 4.

The participating systems were required to perform a three-way classification of tweets: positive, negative or neutral. As the main quality measure we used macro-average F-measure calculated as the average value between F-measure of the positive class and F-measure of the negative class. So we ignored Fneutral because this category is usually not interesting to anybody. But this does not reduce the task to the two-class prediction because erroneous labeling of neutral tweets negatively influences on Fpos and Fneg. Additionally micro-average F-measures were calculated for two sentiment classes.

Table 5. Results of participants in tweet classification tasks. The identifiers of participants in review and Twitter tasks are different

Domain	Measure	Baseline	Participant results	Participant identifier
Telecom	Macro F	0.182	**0.488**	**2**
			0.483	2
			0.480	3
Telecom	Micro F	0.337	**0.536**	**2**
			0.528	10
			0.510	3
Banks	Macro F	0.127	**0.360**	**4**
			0.352	10
			0.335	2
Banks	Micro F	0.238	**0.366**	**2**
			0.364	2
			0.343	8

In Table 5 we present the best results of tweet sentiment analysis for each domain and measure. Most best approaches in this task utilized SVM classification method. The features of the participant 2 comprised syntactic links presented as triples (head word, dependent word, type of relation). Participant 3 applied a rule-based method accounting syntactic relations between sentiment words and the target entities without any machine learning.

Additionally, one of participants fulfilled independent expert labeling of telecom tweets and obtained Macro-F—0.703, and Micro F—0.749, which can be considered as the maximum possible performance of automated systems.

The analysis of the obtained results showed that the most participants solved the general (not entity-oriented) task of tweet classification; entity-oriented approaches did not achieve better results in comparison with general approaches on tweets mentioned several entities.

6. Conclusion

In this paper we described the data, rules and results of SentiRuEval, evaluation of Russian object-oriented sentiment analysis systems. We offered two tasks to participants. The first task was aspect-oriented analysis of reviews about restaurants and automobiles, that is the primary goal was to find word and expressions indicating important characteristics of an entity (aspect terms) and then classify them into polarity classes and aspect categories.

The second task was the reputation-oriented analysis of tweets concerning banks and telecommunications companies. The goal of this analysis was to classify tweets in dependence of their influence on the reputation of the mentioned company. Such tweets could express the user's opinion or a positive or negative fact about the organization.

In each task about ten participants from universities and the industry took part. They have applied various machine-learning approaches including SVM, gradient boosting, CRF, recurrent neural networks and others. Given the participants' results, it can be concluded that the object-oriented sentiment analysis is poorly addressed by the applied methods. And most systems and methods need to be significantly improved to perform better on such tasks.

In the review collections interesting linguistic phenomena were also marked up. In particular, we have labeled comparisons with other entities or with previous opinions, desirable but not existing situations, irony. So the study of the markup can be useful also for linguists. All prepared materials are accessible for research purposes (reviews: http://goo.gl/Wqsqit and tweets: http://goo.gl/qHeAVo).

Acknowledgements

This work is partially supported by RFBR grants No. 14-07-00682, No. 15-07-09306 and by the Russian Ministry of Education and Science, research project No. 586.

References

1. *Amigo E., Corujo A., Gonzalo J., Meij E., de Rijke M.* (2012), Overview of RepLab 2012: Evaluating Online Reputation Management Systems, CLEF 2012 Evaluation Labs and Workshop Notebook Papers, Rome.
2. *Amigo E., Albornoz J. C., Chugur I., Corujo A., Gonzalo J., Martin T., Meij E., de Rijke M, Spina D.* (2013), Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems, CLEF 2013, Lecture Notes in Computer Science Volume 8138, pp. 333–352.
3. *Arora R., Srinivasa S.* (2014), A Faceted Characterization of the Opinion Mining Landscape, COMSNETS Workshop on Science and Engineering of Social Networks, Bangalore, pp. 1–6.
4. *Bagheri A., Saraee M., de Jong F.* (2013), An Unsupervised Aspect Detection Model for Sentiment Analysis of Reviews, in Natural Language Processing and Information Systems, Springer, Berlin, Heidelberg, pp. 140–151.
5. *Barbieri F., Saggion H.* (2014), Modelling Irony in Twitter: Feature Analysis and Evaluation, Proceedings of LREC, pp. 4258–4264.
6. *Chetviorkin I. I., Braslavski P. I., Loukachevitch N. V.* (2012), Sentiment Analysis Track at ROMIP 2011, Proceedings of International Conference Dialog, pp. 739–746.
7. *Chetvirokin I., Loukachevitch N.* (2013), Sentiment Analysis Track at ROMIP 2012, Proceedings of International Conference Dialog, volume 2, pp. 40–50.
8. *Dang H. T., Owczarzak K.* (2008), Overview of the TAC 2008 Opinion Question Answering and Summarization Tasks, Proceedings of the First Text Analysis Conference.
9. *Glavaš G., Korencic D., Šnajder J.* (2013), Aspect-Oriented Opinion Mining from User Reviews in Croatian, Proceedings of the 4th Workshop on Balto-Slavonic Natural Language Processing, pp. 18–22.
10. *Jiang L., Yu M., Zhou M., Liu X., Zhao T.* (2011), Target-dependent Twitter Sentiment Classification, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 151–160.
11. *Jindal N., Liu B.* (2006), Mining Comparative Sentences and Relations, Proceedings of the 21st National Conference on Artificial Intelligence, Boston, pp. 1331–1336.
12. *Kusnetsova E., Loukachevitch N., Chetviorkin I.* (2013), Testing Rules for a Sentiment Analysis System, Proceedings of International Conference Dialog, pp. 71–80.
13. *Liu B.* (2012), Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, Vol. 5(1).
14. *Macdonald C., Santos R., Ounis I., Soboroff I.* (2010), Blog Track Research at TREC, ACM SIGIR Forum, Vol. 44(1), pp. 58–75.
15. *Mohammad S. M., Kiritchenko S., Zhu X.* (2013), NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets, Proceedings of 7th International Workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, pp. 321–327.
16. *Nakov P., Kozareva Z., Ritter A., Rosenthal S., Stoyanov V., Wilson T.* (2013), Semeval-2013 Task 2: Sentiment Analysis in Twitter, Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), Atlanta, pp. 312–320.

17. *Pang B., Lee L., Vaithyanathan S.* (2002), Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Vol. 10, pp. 79–86.
18. *Pontiki M., Galanis D., Pavlopoulos J., Papageorgiou H., Androutsopoulos I., Manandhar S.* (2014), SemEval-2014 Task 4: Aspect Based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 27–35.
19. *Popescu A. M., Etzioni O.* (2005), Extracting Product Features and Opinions from Reviews, Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, pp. 339–346.
20. *Riloff E., Qadir A., Surve P., De Silva L., Gilbert N., Huang R.* (2013), Sarcasm as Contrast between a Positive Sentiment and Negative Situation, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 704–714.
21. *Rosenthal S., Ritter A., Nakov P., Stoyanov V.* (2014), SemEval-2014 Task 9: Sentiment Analysis in Twitter, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 73–80.
22. *Seki Y., Evans D. K., Ku L. W., Sun L., Chen H. H., Kando N.* (2008), Overview of Multilingual Opinion Analysis Task at NTCIR-7, Proceedings of NTCIR-7 Workshop Meeting, Tokyo, pp. 185–203.
23. *Stenetorp P., Pyysalo S., Topić G., Ohta T., Ananiadou S., Tsujii J.* (2012), BRAT: a Web-based Tool for NLP-assisted Text Annotation, Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, pp. 102–107.
24. *Taboada M., Brooke J., Tofiloski M., Voll K., Stede M.* (2011), Lexicon-Based Methods for Sentiment Analysis, Computational Linguistics, Vol. 37(2), pp. 267–307.
25. *Zhang L., Liu, B.* (2014), Aspect and Entity Extraction for Opinion Mining, in Data Mining and Knowledge Discovery for Big Data, Springer, Berlin, Heidelberg, pp. 1–40.

СЕНТИМЕНТНЫЙ АНАЛИЗ ТВИТОВ НА ОСНОВЕ СИНТАКСИЧЕСКИХ СВЯЗЕЙ

Адаскина Ю. В. (adaskina@gmail.com),
Паничева П. В. (ppolin86@gmail.com),
Попов А. М. (hedgeonline@gmail.com)

InfoQubes, Москва, Санкт-Петербург, Россия

Ключевые слова: сентиментный анализ, синтаксические связи, статистические алгоритмы, классификация текстов

SYNTAX-BASED SENTIMENT ANALYSIS OF TWEETS IN RUSSIAN

Adaskina Yu. V. (adaskina@gmail.com),
Panicheva P. V. (ppolin86@gmail.com),
Popov A. M. (hedgeonline@gmail.com)

InfoQubes, Moscow, Saint Petersburg, Russia

The paper describes our approach to the task of sentiment analysis of tweets within SentiRuEval—an open evaluation of sentiment analysis systems for the Russian language. We took part in the task of object-oriented sentiment analysis of Russian tweets concerning two types of organizations: banks and telecommunications companies. On both datasets, the participants were required to perform a three-way classification of tweets: positive, negative or neutral.

We used various statistical methods as basis for our machine learning algorithms and checked which features would provide the best results. Syntactic relations proved to be a crucial feature to any statistical method evaluated, but SVM-based classification performed better than the others. Normalized words are another important feature for the algorithm.

The evaluation revealed that our method proved to be rather successful: we scored the first in three out of four evaluation measures.

Key words: Sentiment analysis, syntactical relations, statistical methods, text classification

Introduction

In spite of being quite well explored by researches and businesses alike sentiment analysis remains to this day one of the most in-demand NLP tasks. Sentiment analysis had been applied on various levels, starting from the whole text level, then going towards the sentence level. Lately most of work has been focused on object-oriented and aspect based sentiment analysis, which is based on the assumption that different opinions can be expressed within one sentence. Today's research dwells not only on the development of automatic sentiment analysis algorithms, but also on evaluation methods. A number of independent bodies conduct evaluations, one of them being Dialogue Evaluation which is held in coordination with Dialogue—the international conference on computational linguisics. This is their third event devoted to sentiment analysis; the results of the first two are discussed in (Chetviorkin, Braslavskiy, Loukachevitch 2012) and (Chetviorkin, Loukachevitch 2013). This year's tasks was automatic evaluation of sentiment towards specific objects or their properties in different datasets (Loukachevitch et al. 2015).

This paper describes our approach to the task. We participated in the object-oriented sentiment analysis of Russian tweets concerning two types of organizations: banks and telecommunications companies. On both datasets, the participants were required to perform a three-way classification of tweets: positive, negative or neutral.

We applied SVM classification (Pedregosa et al. 2011) in our final experiments, although our preliminary results suggested that there was no significant difference between SVM and Naïve Bayes in this task. We used normalized words (further called lemmas) combined with syntactic relations as features. The latter are defined as triplets: source word, target word, relation type. Syntactic relations turned out to be crucial for any statistical method we used in our preliminary tests. All the methods we used showed better results on tweets about telecommunications companies, than on tweets about banks. The evaluation revealed that our method proved to be rather successful: we scored the first in three out of four evaluation measures.

Related work

(Pang, Lee, Vaithyanathan 2002) is generally considered the principal work on using machine learning methods of text classification for sentiment analysis; it explores the use of Naïve Bayes, Maximum Entropy and Support Vector Machines methods. The problem is further discussed in (Go, Bhayani, Huang 2009; Barbosa, Feng 2010 and Jiang et al. 2011), among others. Numerous research was dedicated to developing the ultimate feature set for each specific task to get the best result of automatic classification. Most common features are:

- word forms;
- normalized words;
- phrases;
- frequencies;
- TF-IDF;

- n-gram;
- binary occurrences;
- syntactic relations.

Syntactic information is less common than other parameters because clearly it presupposes a complicated and time-consuming stage of syntactic analysis. However, those experiments that involved dependency relations showed that syntax contributes significantly to both Recall and Precision of most algorithms. For the task of text classification in general see (Furnkranz, Mitchell, Rilof 1998), (Caropreso, Matwin, Sebastiani 2001), (Nastase, Shirabad, Caropreso 2006). (Matsuko et al. 2005) deal with a task very close to ours, sentiment classification based on syntactic relations. They parsed frequent subtrees using two different algorithms, which is a more general approach than ours since we only used 'binary sub-trees', i.e. a pair of words in syntactic relationship. Another distinction is that we combined syntactic information with normal forms as features for machine learning based sentiment classification. (Bethard, Martin 2007) as well as (Zhang et al. 2007) used syntactic relations for the task of semantic relations mining. In (Zhao, Grishman 2005) the authors tackle the task of automatic context extraction, and syntactic relations are a key to their impressive 70% F-measure result.

The sentiment analysis of Twitter today is a full-fledged subtask within sentiment analysis per se. Due to the limited character count the analysis of tweets is closer to sentence-level sentiment analysis than the other blogging platforms. A number of papers discuss the specifics of Twitter sentiment analysis, see for example (Pak, Paroubek 2010; Kouloumpis, Wilson, Moore 2011; Jansen et al. 2009; Tumasjan et al. 2010).

Dataset and task description

We took part in a testing procedure of sentiment analysis systems with our algorithm. Full evaluation details are outlined in (Loukachevitch et al. 2015). The dataset consisted of training and evaluation sets, 10,000 tweets each. Both sets were divided into two subsets: 5,000 tweets about banks and 5,000 tweets about telecommunications companies. The training set had been manually annotated by SentiRuEval experts. This annotation included three-way annotation (negative, positive and neutral) for every company (seven telecommunications companies and eight banks) that was mentioned in the tweet. The test set had been annotated with neutrals for every company that was mentioned in the tweet. Within our task we needed to perform automatic sentiment analysis on the test set, which is either to retain a neutral annotation for the appropriate brand, or to change it to negative annotation or to a positive one. The evaluation set had been annotated by three assessors, and tweets where there was no agreement between the experts (at least two of the three), were excluded from the evaluation set. The total size of the evaluation set was 4,549 tweets for banks and 3,845 tweets for telecommunications companies.

Algorithm

We used InfoQubes morphosyntactic analyzer applied also in (Adaskina, Panicheva, Popov 2014). This is a commercial platform designed by our company. Its lemmatization module is based on Zaliznyak's Grammar Dictionary (Zaliznyak 1980); its syntactic module is a finite state machine, which parses word sequences and produces syntactic trees. An elaborated rule system (featuring 515 syntactic rules) is applied as input context-free grammar for the parser. Every syntactic rule joins two words or phrases into one higher-order phrase and sets respective syntactic relations. Thus, a constituency grammar is applied which in turn yields a dependency structure following a small number of rules. Only binary relations are allowed; each syntactic relation is characterized by three elements: source word, target word and relation type. In total, the system features 19 syntactic relations, their frequencies for both training datasets are presented in Table 1. In our parametrical model the relation (Argument) which has four subtypes (Subject, DirectObject, IndirectObject, PassiveSubject) is split into four different relations.

Table 1. Syntactic relation extracted for the training datasets

Relation Name	Occurrences in Telecom dataset	Occurrences in Banks dataset
Argument:DirectObject	2,778	2,372
Argument:IndirectObject	5,748	3,585
Argument:PassiveSubject	291	232
Argument:Subject	3,148	1,805
Attribute	6,814	6,682
Auxiliary	578	208
Circumstance	3,033	1,211
Coordinate	1,008	1,698
Determiner	687	239
Genitive	3,963	3,355
Identity	2,200	4,937
Infinitive	772	465
Modifier	707	294
Phrasal	1,519	959
Possessive	368	126
Preposition	6,582	4,554
Quantifier	501	605
Subordinate	226	77
Undefined	1,050	1,159

We tested simple word lemmas (unigrams), word lemma bigrams and syntactic relations as features for SVM and Naïve Bayes (Pedregosa et al. 2011) three-way classification (neutral, positive, negative) algorithms. In every experiment we normalized word forms according to the lemmatization module of the morphosyntactic tool. One

of our underlying goals was to test the performance of syntax-based features in the sentiment analysis task. As optional settings we applied a negation marker provided by our morphosyntactic system. Negation marker in our system is a feature that marks cases where a negation particle is connected to the word. We also optionally removed from the parameter list everything that contained words denoting brands in question, implying that an overall brand bias could affect the result negatively. The features and their optional settings are summarized in Table 2.

Table 2. Feature descriptions

Features type	Feature text	Feature type	Options	Example	Comments
1	ВАРИАНТ	Lemma	No negation marker	Lemma *ВАРИАНТ*	Just normalized words
2	ВАРИАНТ|Argument|НЕТ|PassiveSubject	Syntactic relation	No negation marker	Passive subject relation *ВАРИАНТА НЕТ*	Syntactic relation of a certain type between two words. Relation 'Argument' also has four subtypes (Subject, DirectObject, IndirectObject, PassiveSubject), so the subtype is included
3	ВАРИАНТ|Attribute|ЭТОТ|	Syntactic relation	No negation marker	Attribute relation *ЭТОТ ВАРИАНТ*, words are not negated	Syntactic relation of a certain type between two words
4	КРУТОЙ|ВАРИАНТ	Bigram	No negation marker	Bigram *КРУТОЙ ВАРИАНТ*	Two adjacent words
5	ДРУГОЙ|ВАРИАНТ	Bigram	No negation marker	Bigram *ДРУГОЙ ВАРИАНТ*	Two adjacent words
6	ВАРИАНТ|0	Lemma	Negation marker included	Lemma *ВАРИАНТ*, not negated	A combination of normalized words and negation information; here the word is not negated

Features type	Feature text	Feature type	Options	Example	Comments
7	ВАРИАНТ|1	Lemma	Negation marker included	Lemma *ВАРИАНТ*, negated	A combination of normalized words and negation information; here the word is negated
8	ВАРИАНТ|1| Argument| НЕТ|0| PassiveSubject	Syntactic relation	Negation marker included	Passive subject relation *ВАРИАНТА НЕТ*, *ВАРИАНТ* is negated	A combination of syntactic relation and negation information; here one of words is negated
9	ВАРИАНТ|0| Attribute| ЭТОТ|0|	Syntactic relation	Negation marker included	Attribute relation *ЭТОТ ВАРИАНТ*, words are not negated	A combination of syntactic relation and negation information; here neither word is negated
10	КРУТОЙ|0| ВАРИАНТ|0	Bigram	Negation marker included	Bigram *КРУТОЙ ВАРИАНТ*, words are not negated	A combination of bigrams and negation information; here neither word is negated
11	ДРУГОЙ|0| ВАРИАНТ|1	Bigram	Negation marker included	Bigram *ДРУГОЙ ВАРИАНТ*, *ВАРИАНТ* is negated	A combination of bigrams and negation information; here one of words is negated

Preliminary results

We conducted some preliminary experiments applying ten-fold cross-validation to the training dataset only. Our text analysis algorithm consisted of sentiment classification described above and a rule-based algorithm of relevant brand identification. For every document we compiled a list of triplets: document id, brand id, sentiment score. We evaluated the results by computing the overall Precision, Recall and F1-measure over the lists of triplets obtained by text analysis and from the annotated information. Thus we also evaluated the relevant brand identification algorithm and included neutral class performance comparing to the SentiRuEval evaluation scheme. Results we obtained are presented in the following tables, and the highest scores are marked in bold; Table 3 refers to Telecom companies data, Table 4 to Banks data.

Table 3. Preliminary results for Telecom companies data, SVM

Features type	Experiment options		Evaluation		
	Negation marker	Brand name removal	Precision	Recall	F1-measure
Lemmas	–	–	0.7464	0.7482	0.7473
	+	–	0.7549	0.7567	0.7558
	–	+	0.7554	0.7571	0.7563
	+	+	0.7608	0.7625	0.7616
Relations	–	–	0.7275	0.5567	0.6308
	+	–	0.7228	0.5532	0.6267
	–	+	0.7196	0.5470	0.6216
	+	+	0.7215	0.5484	0.6231
Lemmas + relations	–	–	**0.7715**	**0.7734**	**0.7725**
	+	–	0.7692	0.7710	0.7701
	–	+	0.7675	0.7692	0.7684
	+	+	0.7632	0.7648	0.7640
Lemmas + relations, chi-square selection of 5000 best parameters	–	–	0.5865	0.5879	0.5872
Bigrams	–	–	0.7242	0.7077	0.7158
Bigrams + relations	–	–	0.7204	0.7220	0.7212
Bigrams + lemmas	–	–	0.7650	0.7668	0.7659
Bigrams + lemmas + relations	–	–	0.7684	0.7702	0.7693

Table 4. Preliminary results for Banks data, SVM

Features type	Experiment options		Evaluation		
	Negation marker	Brand name removal	Precision	Recall	F1-measure
Lemmas	–	–	0.9046	0.9061	0.9053
	+	–	0.9021	0.9036	0.9029
	–	+	0.9073	0.9087	0.9080
	+	+	0.9032	0.9046	0.9039
Relations	–	–	0.9040	0.8184	0.8591
	+	–	0.9080	0.8220	0.8628
	–	+	0.9040	0.8171	0.8583
	+	+	0.9066	0.8194	0.8608
Lemmas + relations	–	–	0.9059	0.9074	0.9066
	+	–	0.9047	0.9062	0.9055
	–	+	0.9083	0.9097	0.9090
	+	+	**0.9095**	**0.9108**	**0.9101**

Features type	Experiment options		Evaluation		
	Negation marker	Brand name removal	Precision	Recall	F1-measure
Bigrams	–	–	0.8968	0.8949	0.8959
Bigrams + relations	–	–	0.8957	0.8971	0.8964
Bigrams + lemmas	–	–	0.9021	0.9036	0.9029
Bigrams + lemmas + relations	–	–	0.9026	0.9041	0.9033
Lemmas + relations, chi-square selection of 5000 best parameters	–	–	0.8257	0.8269	0.8263

Our preliminary experiments have shown that a combination of lemmas and syntax relations yield the best results for both datasets, while negation and brand name removal options do not considerably affect the performance. That result is consistent with our initial hypothesis that syntactic features should improve the performance. Bigrams and lemmas are almost as good as relations and lemmas. Naïve Bayes classification has confirmed these tendencies with a small decrease in performance. We also tried excluding some features, but the results were unsatisfactory. The tables above include scores for feature selection of 5,000 best parameters, and one can see that this decreased the resulting score rather significantly. Apart from that, we tried tf-idf value, but it also reduced our evaluation metrics. It appears that the data might be too sparse for the weighting factors to work: they probably would have been useful for an experiment with a larger training set where the frequency of each parameter would be higher, and there would be fewer parameters with unique values.

SentiRuEval testing results

For the final experiment within the testing procedure framework we have chosen SVM classification with lemmas and syntactic relations as features, we have also removed brand names from the feature set as an option. We have also performed an out of competition evaluation of the lemmas-based algorithm. Table 5 below represents evaluation results, the numbers in the last column refer to our experiment types ('lemmas', 'lemmas+relations') or the results by other participants (indicated by their number). In italics is our result obtained out of competition. As the main quality measures the evaluation team used two variations of F-measure: F-micro and F-macro, for details see (Loukachevitch et al. 2015). The best result in each category is marked in bold, and, as one can see from the data, our method scored the first in three out of four evaluation measures.

Table 5. Final evaluation results

Domain	Measure	Baseline	Participant results	Participant identifier
Telecom	Macro F	0.182	**0.488**	**lemmas+rels**
			0.483	lemmas+rels, brands removed
			0.480	3
			…	…
			0.469	*lemmas*
			0.465	*lemmas, brands removed*
	Micro F	0.337	**0.536**	**lemmas+rels**
			0.536	lemmas+rels, brands removed
			0.528	10
			…	
			0.512	*lemmas*
			0.514	*lemmas, brands removed*
Banks	Macro F	0.127	**0.360**	**4**
			0.352	10
			0.345	*lemmas*
			0.345	*lemmas, brands removed*
			0.343	lemmas+rels, brands removed
	Micro F	0.238	**0.366**	**lemmas+rels, brands removed**
			0.364	lemmas+rels
			0.363	*lemmas*
			0.362	*lemmas, brands removed*
			0.343	8

There is a notable difference in performance between the preliminary experiments and the testing procedure results, which is naturally justified by a difference in evaluation methods: we have applied F-measure to all the documents in the former case, while in the latter the neutral documents were excluded..

These results are only partially consistent with our preliminary results and our initial hypothesis: on the Telecom dataset the performance of lemmas and relations combined outdoes lemmas only by approx. 2 per cent in micro and in macro F-measures. On the Banks dataset the result is inconclusive: micro F-measure is better by about 0.3 per cent than lemmas and relations combined, but macro F-measure is about 0.2 per cent better with lemmas only. The Banks dataset is also characterized by overall lower performance when the neutral class is not accounted for in the evaluation, contrary to our preliminary experiments yielding higher performance with 'Banks' comparing to 'Telecom'. This fact and the inconsistency of the 'Banks' results distribution (almost the same performance for lemmas and lemmas with relations) suggest that the algorithms applied can't achieve reliable performance with the modest volumes of negative- and positive-class data.

The closest best results in the SentiRuEval scheme were obtained with techniques involving rule-based fact-extraction, MaxEnt and SVM classifiers over various feature sets mostly including word and letter n-grams.

Conclusions

We have applied a syntax-based statistical algorithm to sentiment analysis tasks in two different topics yielding very high performance results comparing to other techniques. We have used straightforward classification features, slightly improving the performance of a simple lemma approach with syntactic relations or not affecting it where the sparsity of data wouldn't allow for reliable high results: the issue that needs to be further addressed. We have used an elaborate morphosyntactic parser, which had proven useful for another semantic task (Adaskina, Panicheva, Popov 2014).

With sparse and modest-sized data SVM appears to be the best classification method; negation or brand-name semantics do not affect the performance much, though we believe that syntactic relations would convey most of the information carried by the negation option. It also appears that the sparsity of data does not allow for effective feature filtering, which could be an option if we boost feature occurrence by, for example, substituting words with semantic classes.

References

1. *Adaskina Yu. V., Panicheva P. V., Popov A. M.* (2014), Semi-Automatic Lexicon Augmenting Based on Syntactic Relations [Poluavtomaticheskoye popolneniye slovarey na osnove sintaksicheskikh svyazey], Proceedings of Internet and Modern Society Conference, Saint Petersburg, pp. 271–276.
2. *Barbosa L., Feng J.* (2010), Robust Sentiment Detection on Twitter from Biased and Noisy Data, Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, pp. 36–44.
3. *Bethard S., Martin J. H.* (2007), CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features, Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), Prague, pp. 245–248.
4. *Caropreso M. F., Matwin S., Sebastiani F. A.* (2006), Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization. Amita G. Chin (ed.), Text Databases and Document Management: Theory and Practice, Idea Group Publishing, pp. 78–102.
5. *Chetviorkin I., Braslavskiy P., Loukachevich N.* (2012), Sentiment Analysis Track at ROMIP 2011, Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2012"], Bekasovo, pp. 1–14.
6. *Chetviorkin I., Loukachevitch N.* (2013), Evaluating Sentiment Analysis Systems in Russian, Proceedings of BSNLP workshop, ACL, Prague, pp. 12–17.

7. *Furnkranz J., Mitchell T. M., Rilof E.* (1998), A Case Study in Using Linguistic Phrases for Text Categorization on The WWW, Proceedings of the AAAI Workshop on Learning for Text Categorization, Madison, US, pp. 5—12.
8. *Go A., Bhayani R., Huang L.* (2009), Twitter Sentiment Classification Using Distant Supervision, Technical report, Stanford.
9. *Jansen, B. J., Zhang, M., Sobel, K., Chowdury, A.* (2009), Twitter power: Tweets as electronic word of mouth, Journal of the American Society for Information Science and Technology, 60(11), pp. 2169–2188.
10. *Jiang L., Yu M., Zhou M., Liu X., Zhao T.* (2011), Target-dependent Twitter Sentiment Classification, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, US, pp. 151–160.
11. *Kouloumpis E., Wilson, T., Moore J.* (2011), Twitter sentiment analysis: The good the bad and the omg! Artificial Intellingence, pp. 538–541.
12. *Loukachevitch N., Blinov P., Kotelnikov E., Rubtsova Ju., Ivanov V., Tutubalina H.* (2015), Sentirueval: Testing Object-Oriented Sentiment Analysis Systems In Russian.
13. *Matsumoto S., Takamura H., Okumura M.* (2005), Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 301–311. Springer, Heidelberg.
14. *Nastase V., Shirabad J. S., Caropreso M. F.* (2006), Using Dependency Relations for Text Classification. In Proceedings of the 19th Canadian Conference on Artificial Intelligence, Quebec City, pp. 12–25.
15. *Pak A., Paroubek P.* (2010), Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of LREC, Valetta, pp. 75–100.
16. *Pang B., Lee L., Vaithyanathan S.* (2002), Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 2002, Vol. 10, Stroudsburg, US, pp. 79–86.
17. *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 É.* (2011), Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research 12(Oct), pp. 2825—2830.
18. *Tumasjan, A., Sprenger, T. O., Sandner, P., Welpe, I.* (2010), Predicting elections with twitter: What 140 characters reveal about political sentiment. Proceedings of ICWSM, Washington, US, pp. 178–185.
19. *Zaliznjak, A. A.* (1980) Grammatical dictionary for Russian language. Rus. jaz, Moscow
20. *Zhang M., G. Zhou, A. Aw* (2008), Exploring Syntactic Structured Features Over Parse Trees for Relation Extraction Using Kernel Methods, Information Processing and Management, vol. 44, issue 2, pp. 687—701.
21. *Zhao S., Grishman R.* (2005), Extracting Relations with Integrated Information Using Kernel Methods, Proceedings of the 43rd Annual Meeting of the ACL, Ann Arbor, US, pp. 419—426.

SEMANTIC SIMILARITY FOR ASPECT-BASED SENTIMENT ANALYSIS

Blinov P. D. (blinoff.pavel@gmail.com)

Kotelnikov E. V. (kotelnikov.ev@gmail.com)

Vyatka State Humanities University, Kirov, Russian Federation

The paper investigates the problem of automatic aspect-based sentiment analysis. Such version is harder to do than general sentiment analysis, but it significantly pushes forward the limits of unstructured text analysis methods. In the beginning previous approaches and works are reviewed. That part also gives data description for train and test collections.

In the second part of the article the methods for main subtasks of aspect-based sentiment analysis are described. The method for explicit aspect term extraction relies on the vector space of distributed representations of words. The term polarity detection method is based on use of pointwise mutual information and semantic similarity measure. Results from SentiRuEval workshop for automobiles and restaurants domains are given. Proposed methods achieved good results in several key subtasks. In aspect term polarity detection task and sentiment analysis of whole review on aspect categories methods showed the best result for both domains. In the aspect term categorization task our method was placed at the second position. And for explicit aspect term extraction the first result obtained for the restaurant domain according to partial match evaluation criteria.

Key words: SentiRuEval, aspect-based sentiment analysis, machine learning, distributed representations of words, semantic similarity

1. Introduction

In the last few years sentiment analysis became an important task in the field of natural language processing. The task is interesting for researchers because of its intricate properties. Business community is attracted by the task because it opens potentially vast opportunity to analyze unstructured text and keep track of target audience attitude to a product or brand.

Formulation of sentiment analysis problem is evolving rapidly with respect to granularity: from whole text and sentences to phrase level (Feldman, 2013). The last level of analysis is the most detailed version that is capable to disentangle complex opinions in reviews. Opinions and sentiments are analyzed with respect to specific aspects of reviewed object, for example, aspects *food*, *service* and *price* of an object *restaurant*. Such detailed task is called aspect-based sentiment analysis (Liu, 2012). For simplification the task can often be split into following subtasks:

1) aspect term extraction;
2) aspect term polarity detection;
3) aspect category polarity detection.

In this article we present new methods for addressing these subtasks. The methods are mainly based on distributed representations of words and notion of semantic similarity.

The rest of the paper is structured as follows. Section 2 gives the overview of previous works. The characteristics of train and test text data are given in Section 3. Section 4 contains method descriptions and results for proposed subtasks. The final conclusions are given in Sections 5.

2. Related work

There are many research papers for sentiment analysis problem, fewer about aspect-based version of it. As for the language, plenty of works were carried out for English (Liu, 2012) and less fewer for Russian (Blinov, Kotelnikov, 2014). Recently there was a burst of research interest to the task because of SemEval-2014 Workshop (Pontiki et al., 2014), where one of the key topics was an aspect-based sentiment analysis. Here we give a brief analysis of applied approaches and methods regarding two main subtasks: aspect term extraction and aspect term polarity detection.

To address aspect term extraction problem participants resorted to two main approaches (Liu, 2012):

1) frequency-based approach;
2) machine learning approach.

Perhaps the first and most famous work from the first approach is (Hu, Liu, 2004). In a nutshell, the general idea of the approach is to find nouns and noun phrases and by some technique filter them out to left only relevant aspect terms. Statistical criteria are often used as such filters (Schouten et al., 2014). Rule-based and dependency parsing methods constitute another group of such filtering techniques (Pekar et al., 2014; Zhang et al., 2014).

The given task can be easily formulated in terms of information extraction tasks, so another popular approach is based on sequence labeling methods. SemEval-2014 Workshop's participants widely used well known Conditional Random Fields (CRF) method (Kiritchenko et al., 2014; Chernyshevich, 2014). In fact the best results in aspect term extraction task were attained by this method with common named entity recognition features and features based on various name lists and word clusters (Toh, Wang, 2014). Each word can be described in terms of features, so traditional machine learning methods for classifications are also used to address the task (Brun et al., 2014; Gupta, Ekbal, 2014).

For the aspect term polarity detection task the most of the solutions exploit external sentiment resources. (Bornebusch et al., 2014) used Stanford sentiment trees to detect terms' sentiments. The best results (Wagner et al., 2014) were obtained by SVM classifier and features based on combination of four rich sentiment lexicons.

3. Text data

This year sentiment analysis evaluation was organized in Russian and was called *SentiRuEval* (Loukachevitch et al., 2015). The evaluation included two types of tasks: aspect-oriented sentiment analysis of users' reviews and object-oriented sentiment analysis of Russian tweets. The article deals with the first of these tasks.

The organizers provide the train data for two domains: restaurant and automobile reviews. Each reviewed object was broken down into several aspects (also referred as aspect categories). For a restaurant there were four aspects: *Food*, *Interior*, *Service* and *Price*. And an automobile was analyzed by six aspects: *Comfort*, *Appearance*, *Reliability*, *Safety*, *Driveability* and *Costs*. In addition each aspect list was supplemented with aspect *Whole* to represent object itself.

The train reviews were manually annotated with mentioned aspect terms according to aspects listed above. There are different types of aspect terms (Loukachevitch et al., 2015), but in our study we focus only on explicit aspect terms. Assessors also were asked to specify sentiment toward terms using four-point scale: *positive*, *negative*, *neutral* and *both*. Thus each aspect term incorporates information about aspect category and polarity. All marked texts were stored in xml format documents. Detailed quantitative characteristics of explicit terms for the train and test data for both domains are given in Table 1. By analyzing the table one can see the usual peculiarity of sentiment analysis tasks: significant skewness toward positive class.

Table 1. Explicit aspect and sentiment distribution

		Number of terms			
		Restaurant		Automobile	
		Absolute	%	Absolute	%
Train	Positive	1,679	69.5	1,513	48.0
	Negative	380	13.5	858	27.2
	Neutral	714	25.3	690	21.9
	Both	49	1.7	91	2.9
	Total	**2,822**	**100**	**3,152**	**100**
Test	Positive	2,478	70.7	1,706	54.9
	Negative	509	14.5	844	27.1
	Neutral	440	12.5	454	14.6
	Both	79	2.3	105	3.4
	Total	**3,506**	**100**	**3,109**	**100**

Besides marked data the organizers provide unlabeled text data for each domain: 19,034 reviews for restaurant domain and 8,271 reviews for automobile domain. All text was preprocessed by morphology analyzer Mystem[1].

[1] Morphological analyzer for Russian mystem. URL: http://tech.yandex.ru/mystem.

4. Aspect-based sentiment analysis

Distributed representations of words show ability to cluster semantically similar words (Mikolov et al., 2013). This property can be useful for solving main subtasks of aspect-based sentiment analysis. In our methods for obtaining distributed representations we use skip-gram model (Mikolov et al., 2013) in the implementation of Gensim library[2]. That model gives us whole vector space in which word vectors are embedded. To produce 300-dimensional word vectors the context window of five words was used. The only texts provided by the organizers were used as the input data for the skip-gram model. But more unlabeled texts lead to better word representations which certainly facilitate performance of proposed method.

4.1. Explicit aspect term extraction method

In the workshop SentiRuEval there were two tasks related to aspect term extraction. Our method deals only with explicit aspect term extraction—task A.

Since the train collection is labeled with aspect terms the initial sets of seed words can be constructed for each aspect. All single-word terms (nouns and verbs) were selected.

For an unknown word-vector $\vec{a} = (a_1, ..., a_n)$ similarity to particular aspect *asp* specified by seed word-vectors $\vec{b}_i = (b_1, ..., b_n)$ can be calculated via cosine similarity in the vector space (Manning et al., 2008):

$$sim(\vec{a}, asp) = \sum_{i=1}^{k} \frac{\vec{a} \cdot \vec{b}_i}{\|\vec{a}\| \cdot \|\vec{b}_i\|}, \vec{b}_i \in B_{asp}, \quad (1)$$

where B_{asp} is the set of seed words for aspect *asp* and $|B_{asp}| = k$ is the number of seed words.

If that similarity exceeds a threshold then the word is marked as aspect term. Thresholds for each aspect category were defined by 10-fold cross validation.

However such procedure can find only single word aspect terms. But multi-word terms form a significant part of all aspect terms, especially for particular aspects, for example *Food*. By our estimate on the restaurant train collection about a fifth part of all terms are multi-word terms. And even greater proportion is preserved for automobile train collection. Probably the multi-word terms can be proceeded naturally by distributed representations but it requires additional preprocessing step to reveal such phrases (with high accuracy) before streaming them to skip-gram model. Very likely it also will require more amount of unlabeled texts. Such improvements lay beyond our current experiments and we resorted to more simple technique to tackle multi-word term issue.

A set of rules was applied to join single terms into a complex one. Sequentially marked words were merged and the ones conjoined by prepositions also merged in a single aspect term. For example, *котлетки из лосося* (*meatballs from salmon*) or *роллы на гриле* (*rolls on grill*). Another set of rules handles aspect terms of category

[2] Topic modeling library gensim. URL: http://radimrehurek.com/gensim.

Whole. Because reviewers often refer to a restaurant by name which is contained in review's metadata, the full match with that string in the text of review is marked as an aspect term.

The baseline method for that task memorizes aspect terms from the train reviews and look for the same terms in the test reviews. Table 2 shows baseline results, best results and results of our method with respect to exact and partial matching evaluation criteria (Loukachevitch et al., 2015). We apply following notion (here and for other tasks' results): **bold** for the best result and *italic* for our method's result. F_1-measure was a primary measure for the tasks.

Table 2. Results of explicit aspect term extraction task (task A)

		Exact matching (macro)			Partial matching (macro)		
	run_id	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
Restaurant	baseline	55.70	69.03	60.84	65.80	69.60	66.51
	2_1	72.37	57.38	**63.19**	80.78	61.65	68.91
	4_1	*55.06*	*69.01*	*60.70*	*68.86*	*79.16*	***72.84***
Automobile	baseline	57.47	62.87	59.41	74.49	67.24	69.66
	2_1	76.00	62.18	**67.61**	85.61	65.51	73.04
	3_1	66.19	65.60	65.13	79.17	72.72	**74.82**
	4_1	*55.77*	*63.55*	*58.63*	*74.17*	*68.87*	*70.16*

Our method shows the best result in term extraction for the restaurant domain according to partial matching, but for exact matching the result is worse. For both variants of evaluation the method shows higher recall values then precision. This means that the method found many terms similar to aspect terms which in fact are not.

For the automobile domain our results are near baseline. This is probably due to small amount of unlabeled additional data. To obtain good vector space one need as much text data as possible. But for the automobile domain additional collection was four times smaller than for restaurant domain. Different aspect term compositionality is another possible explanation of such poor results. For example, in this domain there are mixed terms containing numbers and words such as *Двигатель 2.5 литра* (*The engine of 2.5 liters*), *ваз 2114* (*VAZ 2114*), etc. But our algorithm doesn't take this into account.

In general the baseline benchmarks for each domain are pretty high and even the best participants' results exceed them marginally (all gains are less than 10%). One of the possible reasons of relatively simple applied baseline algorithms' high results (Loukachevitch et al., 2015) is high-quality train collection, which covers a lot of aspect term lexicon which is rather limited.

4.2. Aspect term polarity detection method

The task C was to determine sentiments toward predefined aspect terms. The train examples were classified into four-point scale: *positive*, *negative*, *neutral* and

both. But the evaluation was performed only on three-point scale: *positive*, *negative* and *both*. So we prepared solution to that scale only.

In most cases sentiment of an aspect term is defined by its context words. To represent this context from sentiment perspective sentiment lexicon was created for each domain. All verbs and adjectives are the units of such resource. Only one type of negation (as most common) is handled: <*not*> + <*adjective* or *verb*>. To associate sentiment with each unit we use two types of weighting: based on semantic similarity and based on pointwise mutual information (PMI). The reason of using of two kinds of scores is that two different sources of sentiment information allow better estimate actual sentiment.

For semantic similarity weighting we apply the same procedure for sum similarity calculation (1) for each sentiment unit (represented by real-valued vector $\vec{a}$). The only difference in the task A is the set of words. Now these words are etalon for *positive* or *negative* sentiment. From two sum similarities (to positive and negative classes) the largest by absolute value with appropriate sign became sentiment score for a unit. Examples of such estimation are: *приятный* (+*7.1*) (*nice*); *прекрасный* (+*6.5*) (*lovely*); *стильный* (+*5.9*) (*stylish*); *неуместный* (–*4.8*) (*inappropriate*); *пошлый* (–*4.4*) (*vulgar*); *жуткий* (–*4.2*) (*spooky*); etc.

PMI scores for the same dictionary units were calculated based on collection of reviews with general scores. Collections for PMI calculation previously were filtered out to save most positive (restaurant domain: $score \geq 7 \rightarrow +1$ and automobile domain: $score \geq 4 \rightarrow +1$) and most negative (restaurant and automobile domain: $score \leq 3 \rightarrow -1$) reviews. The score for a unit *w* is defined as (Islam, Inkpen, 2006):

$$score\,(w) = PMI\,(w, pos) - PMI\,(w, neg). \tag{2}$$

Mutual information between unit *w* and, for example, *positive* sentiment class *PMI*(*w*, *pos*) (and for the *negative* class *PMI* was calculated in a similar way) is defined as (Islam, Inkpen, 2006):

$$PMI\big(w, pos\big) = \log_2 \frac{count\big(w, pos\big) \cdot N}{count\big(w\big) \cdot count\big(pos\big)}, \tag{3}$$

where *count*(*w*, *pos*)—count of unit *w* in positive reviews, *N* is total number of tokens in corpus, *count*(*w*)—count of unit *w* in all reviews, *count*(*pos*) is a total amount of terms in positive reviews.

There was no notion of a threshold for PMI scores and each unit of the lexicon assigned to some score. Examples are: *классный* (+*3.1*) (*cool*); *добротный* (+*2.6*) (*mighty*); *выдающийся* (+*1.6*) (*outstanding*); *тошнить* (–*2.7*) (*to puke*); *не дружелюбный* (–*3.8*) (*not friendly*); *хамский* (–*4.5*) (*boorish*); etc.

With the help of weighted dictionary units each aspect term is presented in near (three nearest words) and far (six words) contexts as feature vector. In such form train data is used as an input to gradient boosting classifier (Friedman, 2001).

The sentiment class *both* is presented by very small set of samples (see Table 1). And it is a problem for the classifier to learn such minor-represented class. By observing

"both" aspect terms simple regularity was revealed: for the great number of "both" terms there are "but" conjunction in the sentence. And rule "to assign *both* sentiment to a term if there is a 'but' conjunction in the sentence" was applied to resolve the issue.

The baseline method for this task was a very simple one: to assign a major sentiment for a term based on stats from the train collection (mostly *positive*). Results of baseline, our method and second place participants are given in Table 3.

Table 3. Results of aspect term polarity detection (task C)

		Micro-averaging			Macro-averaging		
	run_id	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
Restaurant	baseline	71.04	71.04	71.04	32.09	25.06	26.71
	4_1	*82.49*	*82.49*	***82.49***	*58.72*	*55.69*	***55.45***
	3_1	66.96	66.96	66.96	32.23	24.30	26.96
Automobile	baseline	61.92	61.92	61.92	29.49	26.85	26.48
	4_1	*74.28*	*74.28*	***74.28***	*57.25*	*56.67*	***56.84***
	1_2	65.31	65.31	65.31	35.63	32.97	34.22

4.3. Aspect term classification method

Goal of task D was to categorize predefined set of terms into aspect categories. Some methods can extract terms and at the same time define its aspect category. In this paper, term categorization task taken out into separate stage.

To solve task D we again resorted to similarity between words. In such meaning this task is opposite to task A. The solution is to compute similarity (1) to seed sets of words and choose aspect category that maximize the similarity. For multi-word term single vector representation can be found by averaging out words of the term (since each word is represented by its vector).

The baseline for that task is identical to baseline in task C: assign most frequent category for a term. With described method our team occupied the second place in this task (Table 4).

Table 4. Results of aspect term categorization (task D)

	run_id	**P**	**R**	**F_1**
Restaurant	baseline	87.42	77.37	79.96
	8_1	89.60	84.14	**86.53**
	4_1	*86.27*	*79.63*	*81.10*
Automobile	baseline	66.72	51.89	56.36
	8_1	68.54	63.55	**65.21**
	4_1	*71.46*	*57.50*	*60.77*

It is interesting that for automobile domain the metrics are much lower than for restaurant domain. Probably it is because the lexicon of automobile review is more intertwined and context dependent. For some terms it is hard to decide to which category it belongs to. For example, *руль* (*steering wheel*) belongs to aspect *Drivability* and *Comfort*; *обзор* (*visibility*) occurs in aspect *Comfort* and *Safety*; etc. And in general number of aspect categories are greater for automobile domain: seven whereas there are only five for restaurants.

4.4. Sentiment analysis of whole review on aspect categories

The task E was to define sentiments about aspect categories. Such sentiments related to the whole review rather than individual aspect terms.

As the solution of polarity detection task is performed in three-point scale the task E is automatically addressed in this scale also. By this point each review has a list of aspect terms with defined sentiment and categories. Following mapping was used to cast sentiments to numbers: +1—*positive*, –1—*negative*, 0—*both*. For each category summation over terms sentiment gives total sentiment of aspect category. If there are no terms for some aspect category it is left with "*absence*" value. If at least one category's term has *both* sentiment the entire category is assign to it.

There were not many participants in this task. Again the baseline is just an assignment of the most frequent sentiment for a particular aspect category. Results are shown in Table 5.

Table 5. Results of sentiment analysis of the whole review on aspect categories (task E)

	run_id	F_1
Restaurant	baseline	27.20
	4_1	***45.82***
	10_1	37.28
Automobile	baseline	23.68
	4_1	***43.90***

The obtained results are the lowest for this task (comparing with other tasks) because of its complexity. The method can be misled by incorrectly extracted aspect term or wrongly detected term's sentiment.

5. Conclusions

We described full stack of methods for main subtasks of aspect-based sentiment analysis. To achieve the best possible results the proposed methods actively use notion of semantic similarity between words, statistical measures and hand-crafted rules.

By partial matching evaluation criteria method for aspect term extraction showed the best results for the restaurant domain among fourteen methods. By exact matching the result is worse but still in the top among participants at the fourth position. The method of polarity term detection showed the best results in both domains among seven runs. For the task of aspect terms' categorization our method was placed at the second position. Also the first place for both domains earned the method for sentiment analysis by aspect categories. From the good results we can conclude that the proposed methods can be used for practical applications to perform detailed sentiment analysis of users' reviews.

Another conclusion that can be drawn is about complexity of sentiment analysis for Russian and English. Actually for one task—exact aspect term extraction—we can compare the results with analogous task from SemEval-2014 (Pontiki et al., 2014). There the best result by F_1 measure for the restaurant domain was 84% while in our competition the best result was only 63%. This leads us to the conclusion that aspect term extraction for Russian is more difficult than for English. The possible sources of the problem are free word order and more complex morphology. To overcome that machine learning methods with more extensive usage of linguistically specific knowledge can probably show the better results for object-oriented sentiment analysis.

Acknowledgements

We want to thank the organizers and assessors for their efforts in running such evaluation workshop. This work is supported by the Russian Ministry of Education and Science, research project No. 586.

References

1. *Blinov P. D., Kotelnikov E. V.* (2014), Using Distributed Representations for Aspect-Based Sentiment Analysis, Proceedings of International Conference Dialog, pp. 739–746.
2. *Bornebusch F., Cancino G., Diepenbeck M., Drechsler R., Djomkam S., Fanseu A., Jalali M., Michael M., Mohsen J., Nitze M., Plump C., Soeken M., Tchambo F., Toni, Ziegler H.* (2014), iTac: Aspect Based Sentiment Analysis using Sentiment Trees and Dictionaries, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 351–355.
3. *Brun C., Popa D., Roux C.* (2014), XRCE: Hybrid Classification for Aspect-based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 838–842.
4. *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), Dublin, pp. 309–313.
5. *Feldman R.* (2013), Techniques and Applications for Sentiment Analysis, Communications of the ACM, Vol. 56, pp. 82–89.

6. *Friedman J.* (2001), Greedy Function Approximation: A Gradient Boosting Machine, The Annals of Statistics, Vol. 29, pp. 1189–1232.
7. *Gupta D., Ekbal A.* (2014), IITP: Supervised Machine Learning for Aspect based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 319–323.
8. *Hu M., Liu B.* (2004), Mining and Summarizing Customer Reviews, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177.
9. *Islam A., Inkpen D.* (2006), Second Order Co-occurrence PMI for Determining the Semantic Similarity of Words, Proceedings of the International Conference on Language Resources and Evaluation, pp. 1033–1038.
10. *Kiritchenko S., Zhu X., Cherry C., Mohammad S.* (2014), NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 437–442.
11. *Liu B.* (2012), Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, Vol. 5(1).
12. *Loukachevitch N. V., Blinov P. D., Kotelnikov E. V., Rubtsova Yu. V., Ivanov V. V., Tutubalina E.* (2015), SentiRuEval: Testing Object-oriented Sentiment Analysis Systems in Russian, Proceedings of International Conference Dialog.
13. *Manning C., Raghavan P., Schütze H.* (2008), Introduction to Information Retrieval, Cambridge University Press., New York.
14. *Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.* (2013), Distributed Representations of Words and Phrases and their Compositionality, Proceedings of NIPS, pp. 3111–3119.
15. *Pekar V., Afzal N., Bohnet B.* (2014), UBham: Lexical Resources and Dependency Parsing for Aspect-Based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 683–687.
16. *Pontiki M., Galanis D., Pavlopoulos J., Papageorgiou H., Androutsopoulos I., Manandhar S.* (2014), SemEval-2014 Task 4: Aspect Based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 27–35.
17. *Schouten K., Frasincar F., Jong F.* (2014), COMMIT-P1WP3: A Co-occurrence Based Approach to Aspect-Level Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 203–207.
18. *Toh Z., Wang W.* (2014), DLIREC: Aspect Term Extraction and Term Polarity Classification System, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 235–240.
19. *Wagner J., Arora P., Cortes S., Barman U., Bogdanova D., Foster J., Tounsi L.* (2014), DCU: Aspect-based Polarity Classification for SemEval Task 4, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 223–229.
20. *Zhang F., Zhang Z., Lan M.* (2014), ECNU: A Combination Method and Multiple Features for Aspect Extraction and Sentiment Polarity Classification, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, pp. 252–258.

ИЗВЛЕЧЕНИЕ АСПЕКТОВ, ТОНАЛЬНОСТИ И КАТЕГОРИЙ АСПЕКТОВ НА ОСНОВАНИИ ОТЗЫВОВ ПОЛЬЗОВАТЕЛЕЙ О РЕСТОРАНАХ И АВТОМОБИЛЯХ

Иванов В. В. (nomemm@gmail.com),
Тутубалина Е. В. (tutubalinaev@gmail.com),
Мингазов Н. Р. (nicrotek547@gmail.com),
Алимова И. С. (alimovaIlseyar@gmail.com)

Казанский Федеральный Университет, Казань, Россия

Ключевые слова: анализ тональности текстов, SentiRuEval, отзывы пользователей, извлечение аспектов, категории аспектов

EXTRACTING ASPECTS, SENTIMENT AND CATEGORIES OF ASPECTS IN USER REVIEWS ABOUT RESTAURANTS AND CARS

Ivanov V. V. (nomemm@gmail.com),
Tutubalina E. V. (tutubalinaev@gmail.com),
Mingazov N. R. (nicrotek547@gmail.com),
Alimova I. S. (alimovaIlseyar@gmail.com)

Kazan Federal University, Kazan, Russia

This paper describes a method for solving aspect-based sentiment analysis tasks in restaurant and car reviews subject domains. These tasks were articulated in the Sentiment Evaluation for Russian (SentiRuEval-2015) initiative. During the SentiRuEval-2015 we focused on three subtasks: extracting explicit aspect terms from user reviews (tasks A), aspect-based sentiment classification (task C) as well as automatic categorization of aspects (task D).

In aspect-based sentiment classification (tasks C and D) we propose two supervised methods based on a Maximum Entropy model and Support Vector Machines (SVM), respectively, that use a set of term frequency features in a context of the aspect term and lexicon-based features. We achieved 40% of macro-averaged F-measure for cars and 40,05% for reviews about restaurants in task C. We achieved 65.2% of macro-averaged F-measure for cars and 86.5% for reviews about restaurants in task D. This method ranked first among 4 teams in both subject domains. The SVM classifier is based on unigram features and pointwise mutual information to calculate category-specific score and associate each aspect with a proper category in a subject domain.

In task A we carefully evaluated performance of a method based on syntactic and statistical features incorporated in a Conditional Random Fields model. Unfortunately, the method did not show any significant improvement over a baseline. However, its results are also presented in the paper.

Key words: aspect-based sentiment analysis, sentirueval, user reviews, aspect extraction, aspect categories

1. Introduction

Over the past decade, opinion mining (also called sentiment analysis) has been an important concern for Natural Language Processing (NLP). Since online reviews significantly influence people's decisions about purchases, sentiment identification has a number of applications, including tracking people's opinions about movies, books, and products, etc.

In this study we describe our approaches for solving a task on sentiment analysis, which was formulated as a separate track in the Sentiment Evaluation for Russian (SentiRuEval-2015) initiative. The SentiRuEval task concerns aspect-based sentiment analysis of user reviews about restaurants and cars. The task consists of several subtasks: aspect extraction (tasks A and B), sentiment classification of explicit aspects (task C), and detection of aspects categories and sentiment summarization of a review (tasks D and E). The primary goal of the SentiRuEval task is to find words and expressions indicating important aspects of a restaurant or a car based on user opinions and to classify them into polarity classes and aspect categories (Loukachevitch et al., 2015).

There have been a large number of research studies in the area of aspect-based sentiment analysis, which are well described in Liu (2012) and Pand and Lee (2008). Traditional approaches in opinion mining are based on extracting high-frequency phrases containing adjectives from manually created lexicons (Turney, 2002; Popescu and Etzioni, 2007). State-of-the-art papers have implemented probabilistic topic models, such as Latent Dirichlet Allocation (LDA), and Conditional Random Field (CRF) for multi-aspect analysis tasks (Moghaddam and Ester, 2012; Choi and Cardie, 2010). Sentiment analysis in English has been explored in depth and there are many well-established methods and general-purpose sentiment lexicons that contain a few thousand terms. However, research studies of sentiment analysis in Russian have been less successful. In 2011–2013 studies have focused on solving a task on sentiment analysis during ROMIP sentiment analysis tracks (Chetviorkin and Loukachevitch, 2013; Kotelnikov and Klekovkina, 2012; Blinov et al., 2013; Frolov et al., 2013).

We use the Conditional Random Fields model applied to the aspect extraction task. In task C for aspect-based sentiment classification we propose a method based on a Maximum Entropy model that uses a set of term frequency features in a context of the aspect term and lexicon-based features. The classifier for aspect category detection is based on a SVM model with a set of category-specific features. We achieved 40% of macro-averaged F-measure for cars and 40,05% for reviews about restaurants in task C. We achieved 65.2% of F-measure for cars and 86.5% for reviews about restaurants in task D.

The rest of the paper is organized as follows. In Section 2 we introduce related work on sentiment analysis. In Section 3 we describe proposed approaches. Section 4 presents results of experiments. Finally, in Section 5 we discuss the results.

2. Related Work

In this paper, we focus on the detection of the three major cores in a review: aspect terms, sentiment about these aspects, and aspects' categories. During the last decade, a large number of methods were proposed to identify these elements.

Aspect term extraction. There are several widely used methods that treat the task as a classification problem (Popescu et al., 2005), as a sequence labeling problem (Jakob and Gurevych, 2010; Kiritchenko et al., 2014; Chernyshevich, 2014), as a topic modeling or a traditional clustering task (Moghaddam and Ester, 2012; Zhao et al., 2014). The classification problem is to determine whether nouns and noun phrases are target of an opinion or not. Popescu et al. (2005) used syntactic patterns in relation with sentiment from general-purpose lexicons to identify high-frequency noun phrases. Poria et al. (2014) proposed a rule-based approach, based on knowledge and sentence dependency trees. These approaches are limited due to lower results on extracting low-frequency aspects or hand-crafted dependency rules for complex extraction. In (Kiritchenko et al., 2014; Chernyshevich, 2014) the authors proposed two modifications of a standard scheme for sequence labeling models.

Aspect term polarity. Most of the early approaches for classifying aspects rely on seed words or a manually generated lexicon that contains strongly positive or strongly negative words. Turney (2002) proposed an unsupervised method, based on a sentiment score of each phrase that is calculated as the mutual information between the phrase and two seed words. Recent papers have widely applied machine learning methods to solve the tasks of sentiment classification (Pang et al., 2002; Pang and Lee, 2008; Blinov et al., 2013; Kiritchenko et al., 2014). Moghaddam and Ester (2012) proposed extensions of the LDA model to extract aspects and their sentiment ratings by considering the dependency between aspects and their sentiment polarities. However, topic models achieve lower performance on multi-aspect sentence classification than the SVM classifier in three different domains (Lu et al., 2011).

Aspect category detection. Automatic categorization of explicit aspects into aspect categories has been studied as the task of sentiment summarization. Moghaddam and Ester (2012) investigated it as a part of a latent aspect mining problem. There have been some works on grouping aspect terms from review texts for the sentiment analysis in the task 4 of the international workshop on Semantic Evaluation (SemEval-2014). The task was evaluated with the F-measure and the best results were achieved by SVM classifies with bag-of-words features and information from unlabeled reviews (Pontiki et al., 2014; Kiritchenko et al., 2014).

Several studies about sentiment analysis have been done in Russian, related to evaluation events of Russian sentiment analysis systems (Chetviorkin and Loukachevitch, 2013). Frolov et al. (2013) proposed a dictionary-based approach with fact semantic filters for sentiment analysis of user reviews about books. Blinov et al.

(2013) showed benefits of machine learning method over lexical approach for user reviews in Russian and used manual emotional dictionaries.

3. System description

In this section we describe our approaches for three tasks of aspect-based sentiment analysis of user reviews about restaurants and cars. The CRF model was used for automatic extraction of explicit aspects (task A). We applied machine-learning approaches for the tasks C and D, based on bag-of-words model and a set of lexicon-based features that are described in Section 3.2 and 3.3, respectively. The morpho-syntactic analyzer Mystem was used for text normalization at the preprocessing step.

3.1. Aspect Extraction

The goal of aspect extraction is to detect extract major explicit aspects of a product (task A). Since the task can be seen as a particular instance of the sequence-labeling problem, we employ Conditional Random Fields (Lafferty et al., 2001).

Explicit aspects denote some part or characteristics of a described object such as *передний привод* (*front-wheel drive*), *руль* (*steering wheel*), *динамика* (*dynamics*) in cars reviews; *столик* (*table*), *официант* (*waiter*), *блюдо* (*dish*) in reviews about restaurants. In the following examples we consider user phrases about explicit aspects.

We use Inside-Outside-Begin scheme and Passive Aggressive algorithm for training CRF; brief description of the features used to represent the current token w_i are presented below: the current token w_i, the current token w_i within a window ($w_{i-2}, \ldots, w_{i+2}$); the part of speech tag of the current token; the part of speech tag of the token within a tag window ($tag_{i-2}, \ldots, tag_{i+2}$); the number of occurrences of the tokens in the training set; the presence of the token in manually created domain-dependent dictionaries.

3.2. Aspect-based sentiment classification

The task of sentiment classification aims to predict polarity (positive, negative, neutral, or both) of each aspect from the product reviews. We applied the Maximum Entropy classifier with default parameters, based on a bag-of-words model and a set of lexicon-based features that are described in Section 3.2.2.

The following examples illustrate the aspects (marked in italic) with different polarities from the reviews. Some phrases like "*персонал* улыбчивый, приветливый." ("smiling, friendly staff"), "общее впечатление: отличная *машина*" ("overall impression: great car") or "просторный *салон*, *удобно* сидеть пассажиру сзади" ("*spacious interior*, a passenger could sit *comfortably* behind the driver's seat") contain strong positive or negative context near the aspect term. Therefore, such cases could

be correctly classified extracting bigrams in the phrases. Complex analysis of sentiment phrases such as "заказывал *бифштекс*, нет слов как *вкусно*" ("I ordered a beefsteak, there are no words to describe just how *tasty* this was") and "в городском цикле *компьютер* будет показывать очень неприятные цифры" ("in the city the *computer* will show very unpleasant figures") shows that there is a distance between the polarity words *вкусно (tasty), неприятные* (*unpleasant*) and the aspect terms. We use combinations of the aspect term and a context term to classify these cases. Difficult phrases with both sentiments such as "отмечу некоторую *жесткость сидений*, но привыкаешь, главное сидеть удобно" ("I note some *rigidity of the seats*, but you get used to it, the main thing is sit conveniently") or "*горячее* неплохое, но на гриль было непохоже" ("*hot dishes* are quite good, but not similar to a grill") could be recognized by presence of the conjunction word *но* (*but*).

Given a context of the aspect term, two types of word bigrams are generated for feature extraction: (i) context bigrams, using a text within a context window of the aspect term; (ii) aspect-based bigrams as a combination of the aspect term itself and a context word within the context window. The context window of the aspect term w_i denotes a sequence $(w_{i-4}, \ldots, w_{i+4})$.

3.2.1. Manually created sentiment lexicon

We collected user rated reviews from otzovik.com: 7,526 reviews about restaurants and 4,952 reviews about cars. To make corpus more accurate, we included only *Pros* reviews with an overall rating 5 into positive corpus and *Cons* reviews with an overall rating 1 or 2 into negative corpus. *Pros (Преимущества)* and *Cons (Недостатки)* are parts of a review that describe strong reasons why an author of the review likes or dislikes the product, respectively. For each domain we selected the top K adverbs, adjectives, verbs, reducing noun words that express aspects, action verbs and most common adjectives. The manually created dictionary consists of about 741 positive and 362 negative words in restaurants domain and includes 1,576 positive and 741 negative words in cars domain. We combine two dictionaries to achieve better evaluation results.

For lexicon-based features we use the following scores: each word in the sentence is weighted by its distance from the given aspect:

$$score(w) = \frac{sc(w)}{e^{|i-j|}}$$

where i, j is the positions of the aspect term and the word, $sc(w)$ is the sentiment word's score, that equals 1 for positive words and -1 for negative words, extracted from the sentiment dictionary.

3.2.2. Classification Features for Aspect Term Polarity

Each review is represented as a feature vector, for each aspect features are extracted from the aspect and its context in a sentence. A brief description of the features that we use is presented below:

- **character n-grams**: lowercased characters n-grams for $n = 2, \ldots, 4$ with document frequency greater than two were considered for feature selection.

- **lexicon-based unigrams**: unigrams from the sentiment lexicon are extracted for feature selection.
- **context n-grams:** unigrams (single words) and bigrams are extracted from the context window. We extract these n-grams for several combinations: (i) replacement of the aspect term with the word *aspect*; (ii) replacement of sentiment words with the polarity word *pos* or *neg*; (iii) replacement of sentiment words with a part of speech tag.
- **aspect-based bigrams:** bigrams generated as a combination of the aspect term itself and a word within the context window. We extract these bigrams for several combinations that described above.
- **lexicon-based features:** the features are calculated as follows: the maximal sentiment score; the minimum sentiment score; the sum of the words' sentiment scores; the sum of positive words' scores; the sum of negative words' scores. Sentiment words with negations shift the sentiment score towards the opposite polarity.

Due to limited size of the context window and difficulty in classifying the aspect with both negative and positive sentiment towards its term, we create hand-crafted rule for such cases: if the sentence *(s)* contains the aspect term, a conjunction word *но, а* (*but*) and the classifier predicts the neutral label for the aspect, we mark the aspect by the both label.

3.3. Automatic categorization of explicit aspects into aspect categories

The goal of task D is to classify each aspect to one of predefined categories. In restaurant reviews there are the following aspect categories: *food, service, interior, price, general*. For automobiles aspect categories are: *drivability, reliability, safety, appearance, comfort, costs, general.*

We describe the task of automatic categorization of explicit aspects in the following examples. Some aspects such as food products (e.g., *бифштекс* (*beefsteak*), *утка по-пекински* (*Peking duck*)) or car components (e.g., *гидроусилитель* (*power steering*), *двигатель* (*engine*)) are classified by a human annotator's explicit knowledge. The categories of food products and car components are *food* and *drivability,* respectively. The category label of some explicit aspects depends on a context of a user review. In the examples "*машина* свои деньги отработала полностью" ("the car is worth its price"), "пробовал отпускать руль *машина* едет ровно" ("have experimented with the driving wheel and the car running smoothly"), "*машина* предназначена для фанатов" ("the *car* is intended for fans") and "довольно красивая *машина*" ("quite beautiful *car*") the categories of the aspect term *машина* (*car*) are *costs*, *drivability*, *whole*, *appearance,* respectively.

We addressed the task as a text classification problem and trained the SVM classifier with the sequential minimal optimization (SMO). For each aspect term w_i we extracted the aspect term itself and the features from the context window $(w_{i-2}, \ldots, w_{i+2})$. Category-specific lexicons are based on a score for each term w in the training test:

$$score\,(w) = PMI\,(w, cat) - PMI\,(w, oth)$$

where *PMI* is pointwise mutual information, *cat* denotes all aspects' contexts in the particular category, *oth* denotes aspects' contexts in other categories.

The SVM classifier is based on bag-of-words model and other features described below:

- word n-grams: the aspect term and unigrams from the context of the aspect term are extracted for feature selection.
- category-specific features: the following features are calculated separately for each category: the maximal score in the context; the minimum score in the context; the sum of the words' scores in the context; the average of the words' scores in the context;

4. Experimental Results

For experimental purposes we used the training set of 200 annotated reviews and the testing set of 200 reviews for each domain provided by the organizers of the SentiRuEval task.

4.1. Performance results

The official results obtained by our approaches on the testing set are presented in Tables 1, 2a, 2b and 3. The tables show the official baseline results and the results of other participants according to macro-average F-measure as the main quality measure in the task (Loukachevitch et al., 2015).

For task A exact matching and partial matching were used to calculate F1-measure. Table 1a and 1b show that our method based on the CRF model did not have any significant improvement over a baseline.

For task C macro-averaged F-measure is calculated as the average value between F-measure of the positive class, negative class and F-measure of the both class. Tables 2a show that according to macro-averaged F1-measure, our classifier does not pay off when compared with the approach with run_id 4_1, that is based on a Gradient Boosting Classifier model. Our approach has 0.13% and 0.06% improvements in macro-averaged F1-measure over the approach with run_id 3_1, ranked second in restaurants and banks domain, respectively. Our runs could not be evaluated due to technical problems with the submission.

Table 3 shows the official baseline results and the results of the method, ranked second according to macro-averaged F-measure in task D. This method ranked first among 4 teams in both subject domains. The best approach has 0.06% and 0.09% improvements in macro F1-measure over the baseline in restaurants and cars domains, respectively.

Table 1a. Performance metrics in extraction of explicit aspects in restaurants domain (task A)

	Exact matching			Partial matching		
	Macro P	**Macro R**	**Macro F**	**Macro P**	**Macro R**	**Macro F**
Our method	0.3515	0.5331	0.5331	0.6507	0.4399	0.5109
An approach, ranked first	0.5506	0.6901	0.6070	0.6886	0.7916	0.7284
Official baseline	0.5570	0.6903	0.6084	0.6580	0.6960	0.6651

Table 1b. Performance metrics in extraction of explicit aspects in cars domain (task A)

	Exact matching			Partial matching		
	Macro P	**Macro R**	**Macro F**	**Macro P**	**Macro R**	**Macro F**
Our method	0.6411	0.5363	0.5749	0.7264	0.6117	0.6498
An approach, ranked first	0.6619	0.6560	0.6513	0.7917	0.7272	0.7482
Official baseline	0.5747	0.6287	0.5941	0.7449	0.6720	0.6966

Table 2a. Performance metrics in the classification task in restaurants domain (task C)

Run_id	**Micro P**	**Micro R**	**Micro F**	**Macro P**	**Macro R**	**Macro F**
Official baseline	0.7104	0.7104	0.7104	0.3209	0.2506	0.2671
1_1	0.6194	0.6194	0.6194	0.2517	0.2454	0.2379
1_2	0.6194	0.6194	0.6194	0.2517	0.2454	0.2379
3_1	0.6696	0.6696	0.6696	0.3223	0.2430	0.2696
4_1	0.8249	0.8249	0.8249	0.5872	0.5569	0.5545
Our approach	0.7671	0.7671	0.7671	0.4582	0.3729	0.4081

Table 2b. Performance metrics in the classification task in cars domain (task C)

Run_id	**Micro P**	**Micro R**	**Micro F**	**Macro P**	**Macro R**	**Macro F**
Official baseline	0.6192	0.6192	0.6192	0.2949	0.2685	0.2648
1_1	0.6471	0.6471	0.6471	0.3399	0.3194	0.3293
1_2	0.6531	0.6531	0.6531	0.3563	0.3297	0.3422
3_1	0.5589	0.5589	0.5589	0.3016	0.2621	0.2794
4_1	0.7428	0.7428	0.7428	0.5725	0.5667	0.5684
1_3	0.6252	0.6252	0.6252	0.3507	0.3262	0.3345
Our approach	0.7110	0.7111	0.7111	0.4481	0.3761	0.4001

Table 3. Performance metrics in categorization of aspects in both subject domains (task D)

	Restaurants			Cars		
	Macro P	**Macro R**	**Macro F**	**Macro P**	**Macro R**	**Macro F**
Our approach	0.8960	0.8414	0.8653	0.6854	0.6355	0.6521
Second result	0.8627	0.7963	0.8110	0.7146	0.5750	0.6077
Official baseline	0.8742	0.7737	0.7996	0.6672	0.5190	0.5636

4.2. Ablation Experiments

We performed ablation experiments to study the benefits of features, which are used for the CRF model and machine learning methods. Tables 4a, 4b and 5 show ablation experiments for tasks A and C on the testing set, removing one each individual feature category from the full set. Error analysis and Tables 4a and 4b show that the features on the set of two previous and two next tokens decrease our results in task A in restaurants domain. The most effective features for task C are based on aspect-based bigrams that include combinations of the aspect term and other words from the context window.

Table 4a. Results for the ablation experiments in aspect extraction about restaurants (task A)

	Exact matching			Partial matching		
	P	**R**	**F1**	**P**	**R**	**F1**
all features	0.3515	0.5331	0.5331	0.6507	0.4399	0.5109
w/o dictionaries	0.3382	0.4971	0.3961	0.3850	0.6921	0.4821
w/o frequencies	0.6503	0.4322	0.5068	0.7313	0.4755	0.5612
w/o all tokens within $(w_{i-2}, \ldots, w_i)$	0.6105	0.4065	0.4751	0.7118	0.4667	0.5471
w/o all tokens within $(w_i, \ldots, w_{i+2})$	0.6471	0.4375	0.5104	0.7272	0.4865	0.5681
w/o tokens that contained all features within $(w_{i-1}, \ldots, w_{i+1})$	0.7311	0.4801	0.5644	0.6476	0.4416	0.5120

Table 4b. Results for the ablation experiments in aspect extraction about cars (task A)

	Exact matching			Partial matching		
	P	**R**	**F1**	**P**	**R**	**F1**
all features	0.6411	0.5363	0.5749	0.7264	0.6117	0.6498
w/o dictionaries	0.6451	0.5421	0.5798	0.7303	0.6191	0.6556
w/o frequencies	0.6380	0.5364	0.5742	0.7148	0.6121	0.6455
w/o all tokens within $(w_{i-2}, \ldots, w_i)$	0.6281	0.5217	0.5609	0.7341	0.6077	0.6498
w/o all tokens within $(w_i, \ldots, w_{i+2})$	0.6144	0.5328	0.5624	0.7022	0.6197	0.6453
w/o tokens that contained all features within $(w_{i-1}, \ldots, w_{i+1})$	0.6414	0.5356	0.5742	0.7264	0.6091	0.6472

Table 5. Results for the ablation experiments in sentiment classification towards aspects (task C)

	Restaurants			Cars		
	macro P	**macro R**	**macro F**	**macro P**	**macro R**	**macro F**
All features	0.4582	0.3729	0.4081	0.4481	0.3761	0.4001
w/o character n-grams	0.4479	0.3659	0.4000	0.4480	0.3750	0.3994
w/o lexicon-based unigrams	0.4259	0.3651	0.3921	0.4213	0.3669	0.3869
w/o aspect-based bigrams	0.4261	0.3396	0.3728	0.4380	0.3746	0.3951
w/o context n-grams	0.4355	0.3586	0.3906	0.4370	0.3717	0.3941
w/o lexicon-based scores	0.4629	0.3681	0.4050	0.4374	0.3747	0.3959

Table 6. Results for feature ablation experiments in categorization of aspects (task D)

Combinations of features	Restaurants			Cars		
	P	**R**	**F**	**P**	**R**	**F**
word n-grams	0.7650	0.7193	0.7388	0.6554	0.6060	0.6219
word n-grams + single cumulative score	0.8185	0.7705	0.7914	0.6800	0.6296	0.6461
word n-grams + domain-specific scores	0.8960	0.8414	0.8653	0.6854	0.6355	0.6521

The experiments for task D are presented in Table 6. Through these feature ablation experiments we show that most important features are the domain-specific features, that are based on pointwise mutual information for the category and include four different calculations of scores in the context of the aspect term.

5. Conclusion

In this paper we described supervised methods for sentiment analysis of user reviews about restaurants and cars. In extraction of explicit aspects (task A) we proposed the method based on syntactic and statistical features incorporated in the Conditional Random Fields model. The method did not show any significant improvement over the official baseline. In extraction of sentiments towards explicit aspects (task C) our method was based on the Maximum Entropy model on a set of lexicon-based features and two types of term frequency features: context n-grams and aspect-based bigrams. We demonstrated that by using these features, classification performance increases from baseline macro-averaged F-measures of 0.267 to 0.408 for restaurants and of 0.265 to 0.4 for cars. In categorization of explicit aspects into aspect categories (task D) we proposed the SVM classifier, based on unigram features and pointwise mutual information to calculate category-specific score. We achieved 65.2% of macro-averaged F-measure for cars and 86.5% for reviews about restaurants in task D. This method ranked first among 4 teams in both subject domains. For future work we plan to provide error analysis of the described methods.

Acknowledgments

This work was funded by the subsidy of the Russian Government to support the Program of competitive growth of Kazan Federal University and supported by Russian Foundation for Basic Research (RFBR Project 13-07-00773).

References

1. *Blinov P., Klekovkina M., Kotelnikov E., Pestov O.* (2013), Research of lexical approach and learning methods for sentiment analysis, Computational Linguistics and Intellectual Technologies, Vol. 2(12), pp. 48–58.
2. *Chernyshevich M.* (2014), IHS R&D Belarus: Cross-domain Extraction of Product Features using Conditional Random Fields, SemEval 2014, pp. 309–313.
3. *Chetviorkin I., Loukachevitch N.* (2013), Evaluating Sentiment Analysis Systems in Russian, ACL 2013, p. 14.
4. *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.
5. *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.

6. *Frolov A. V., Polyakov P. Yu., Pleshko V. V.* (2013), Using semantic filters in application to book reviews sentiment analysis, available at: www.dialog-21.ru/digests/dialog2013/materials/pdf/FrolovAV.pdf
7. *Kiritchenko S., Zhu X., Cherry C., Mohammad S. M.* (2014), NRC-Canada-2014: Detecting aspects and sentiment in customer reviews, SemEval 2014, pp. 437–442.
8. *Lafferty J., McCallum A., Pereira F. C.* (2001), Conditional random fields: Probabilistic models for segmenting and labeling sequence data, Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), pp. 282–289.
9. *Liu B.* (2012), Sentiment analysis and opinion mining, Synthesis Lectures on Human Language Technologies, vol. 5(1), pp. 1–167.
10. *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-2015, pp. 3–9.
11. *Lu B., Ott M., Cardie C., Tsou B. K.* (2011), Multi-aspect sentiment analysis with topic models, Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference, pp. 81–88.
12. *Moghaddam S., Ester M.* (2012), On the design of LDA models for aspect-based opinion mining, Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 803–812.
13. *Pang B., Lee L., Vaithyanathan S.* (2002), Thumbs up?: sentiment classification using machine learning techniques, Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol. 10, pp. 79–86.
14. *Pang B., Lee L.* (2008), Opinion mining and sentiment analysis, Foundations and trends in information retrieval, vol. 2(1–2), pp. 1–135.
15. *Pontiki M., Papageorgiou H., Galanis D., Androutsopoulos I., Pavlopoulos J., Manandhar S.* (2014), Semeval-2014 task 4: Aspect based sentiment analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27–35.
16. *Popescu A. M., Etzioni O.* (2007), Extracting product features and opinions from reviews, Natural language processing and text mining, pp. 9–28.
17. *Poria S., Cambria E., Ku L. W., Gui C., Gelbukh A.* (2014), A rule-based approach to aspect extraction from product reviews, SocialNLP 2014, pp. 28–37.
18. *Turney P. D.* (2002), Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, Proceedings of the 40th annual meeting on association for computational linguistics, pp. 417–424.
19. *Zhao Y., Qin B., Liu T.* (2014), Clustering Product Aspects Using Two Effective Aspect Relations for Opinion Mining, Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 120–130.

ВЫСОКОТОЧНЫЙ МЕТОД ИЗВЛЕЧЕНИЯ АСПЕКТНЫХ ТЕРМИНОВ ДЛЯ РУССКОГО ЯЗЫКА

Майоров В. (vmayorov@ispras.ru),
Аванесов В. (avanesov@ispras.ru),
Андрианов И. (ivan.andrianov@ispras.ru),
Астраханцев Н. (astrakhantsev@ispras.ru),
Козлов И. (kozlov-ilya@ispras.ru),
Турдаков Д. (turdakov@ispras.ru)

Институт Системного Программирования РАН, Москва, Россия

Ключевые слова: извлечение аспектных терминов, анализ эмоциональной окраски, извлечение именованных сущностей, автоматическое извлечение терминов

A HIGH PRECISION METHOD FOR ASPECT EXTRACTION IN RUSSIAN

Mayorov V. (vmayorov@ispras.ru),
Andrianov I. (ivan.andrianov@ispras.ru),
Astrakhantsev N. (astrakhantsev@ispras.ru),
Avanesov V. (avanesov@ispras.ru),
Kozlov I. (kozlov-ilya@ispras.ru),
Turdakov D. (turdakov@ispras.ru)

Institute for System Programming of RAS, Moscow, Russia

This paper presents a work carried out by ISPRAS on aspect extraction task at SentiRuEval 2015. Our team submitted one run for Task A and Task B and got best precision for both tasks for all domains among all participants. Our method also showed the best F1-measure for exact aspect term matching for task A for automobile domain and both for Task A and Task B for restaurant domain.

The method is based on sequential classification of tokens with SVM. It uses local, global, syntactic-based, GloVe, topic modeling and automatic term recognition features. In this paper we also present evaluation of significance of different feature groups for the task.

Key words: Aspect Extraction, Sentiment Analysis, NERC, Syntax Trees, Topic modeling, GloVe, Automatic Term Recognition

Introduction

This paper describes participation in aspect extraction tasks of SentiRuEval 2015, which focuses on detecting aspect terms in reviews for restaurant and cars.

Aspect extraction is a part of object-oriented sentiment analysis. An author of a text can have different opinions relative to specific properties of an object called aspects. Aspect terms represent these aspects in particular text.

Organizers of the competition divided all aspect terms into three types: Explicit aspects, Implicit aspects, Sentiment facts (Lukashevich N. V. et. al. 2015). According to the task definition, «Explicit aspects denote some part or characteristics of a described object such as staff, pasta, music in restaurant reviews. [...] Implicit aspects are single words or single words with sentiment operators that contain within themselves as specific sentiments as the clear indication to the aspect category. In restaurant reviews the frequent implicit aspects are such words as tasty (positive+food) [...] Sentiment facts do not mention the user sentiment directly, formally they inform us only about a real fact, however, this fact conveys us a user's sentiment as well as the aspect category it related to. For example, sentiment fact *отвечала на все вопросы* (*answered all questions*) means positive characterization of the restaurant service".

SentiRuEval dataset was annotated with these three subtypes of aspect terms and participants were asked to extract separately only explicit aspect terms and all aspect terms. In the rest of the paper we will refer to explicit aspect extraction task as "Task A" and all aspect extraction task as "Task B".

Our aspect extraction system uses supervised machine learning with support vector machines (SVM) to classify each token of a review into classes which denote beginning or middle of an aspects or term outside aspect. We train our classifier only on explicit aspect terms in order to perform Task A, and use union of results of three different classifiers trained for extraction of each type of aspects separately.

Main challenge was search of good feature space. We define three groups of features: local features computed in the bounds of one sentence; global features calculated for one document; and features that use external resources.

The paper is organized as follows: Section 1 gives brief overview of the related work; in Section 2 we present full description of our method and feature space it uses; Section 3 provides evaluation for different combination of features for each task; in the final section we make conclusion for this work.

1. Related work

Aspect extraction task has been widely studied in recent years. There are four main approaches (Liu, 2012) for this task. The first approach is to extract frequent nouns and noun phrases (Hu & Liu, 2004) (Popescu & Etzioni, 2007) (Scaffidi et al., 2007). The second one utilizes opinion word and target relations (Hu & Liu, 2004) (Qiu et al., 2011) (Poria et al. 2014). These methods are based on the idea that opinion words (i.e. words or phrases that specify sentiment) are related to aspect expressions in reviews. The third approach uses topic modeling (Mei et al., 2007) (Branavan et al.,

2008) (Li, Huang & Zhu, 2010). The last approach is based on supervised machine learning. The most effective methods were shown to be sequential learning, namely Hidden Markov Models (Jin & Ho, 2009) and Conditional Random Fields (Jakob & Gurevych, 2010) (Choi & Cardie, 2010).

2. Method description

2.1. Overview

User's opinion could be expressed in several ways. Each aspect in datasets provided by organizers was marked with one of five types of expression: *relevant* (aspect term mention is relevant for current review object), *comparison* (aspect term is mentioned in comparison with another object), *previous* (aspect term is mentioned in comparison with previous experience), *irrealis* (aspect term is mentioned to describe hypothetical not materialized state of things) and *irony* (aspect term is mentioned with irony). We merged all marks except *relevant* to one class *"other"* due to relatively small number of aspects with marks *comparison*, *irony* etc.

At first we tokenize all reviews and transform task into sequence labelling task: given list of tokens assign sequence of tags to each element of sequence. Our method assigns one of five following classes to each token:

1. Out of aspect term
2. Beginning of *relevant* aspect term
3. Middle of *relevant* aspect term
4. Beginning of *other* aspect term
5. Middle of *other* aspect term

Each token is classified using SVM with L2 regularization. Used features are briefly described below.

We use Texterra system (Turdakov et. al., 2014) as general NLP tasks solution for text tokenization, PoS tagging and morphological analysis. Also we use MaltParser (Nivre et al., 2007) trained on SynTagRus[1] corpora for syntactic parsing.

2.2. Local features

Local features are features that are computed using only sentence. The main local feature used in our method is classification labels of tokens in left window of size 2.

We note that aspect extraction task is very similar to named entity recognition task (NERC). So, we use some features that are successfully used in supervised machine learning NERC method (Zhang & Johnson, 2003). Used NERC features are described in section 2.2.1.

[1] http://www.ruscorpora.ru/instruction-syntax.html

Because Russian language has free word order, we decided to use sentence syntactic structure based features (see section 2.2.2).

2.2.1. NERC features

We note that aspect extraction task is very similar to named entity recognition task. So, as basic features we choose following features that are described in (Zhang & Johnson, 2003).

Token prefixes and suffixes of length 1–4; token word forms, POS tags, morphological properties, lemmas in sentence window of size 2; whether a token placed at start of a sentence; token mask (all digits in token are replaced to a special character) and some token spelling features in window of size 2 (are all characters in uppercase / digits or punctuation marks / non letters / digits or letters; is any character a digit; is first character in uppercase).

2.2.2. Syntactic features

We use following features based on sentence syntactic structure. Distance in sentence syntactic tree between current token and other tokens in window of size 3. Lemma, POS tag and token morphological properties for parent token (in terms of syntactic tree) and for each child token. Classification labels assigned to parent and children tokens in left window.

2.3. Global features

Global features are features that are computed using the whole document. We use some of features used for supervised machine learning based NERC method (Ratinov & Roth, 2009): relative frequency of classification labels for all tokens having an equal word form with current one in left window of size 1000; relative frequency of having upper case first character for all tokens having an equal word form with current one in left window of size 200; relative frequency of POS tags, morphological properties and lemmas for all tokens having an equal word form with current one in left window of size 200.

2.4. Features based on external resources

2.4.1. Glove

We also use word to vector space embedding as features. In order to obtain the embedding to 50-dimensional vector space we train GloVe (Pennington, 2014) on Russian Wikipedia. Unfortunately, the vectors assigned to words are non-interpretable but they are known to be similar (in terms of Euclidean distance) for similar words. In order to obtain interpretable features we discover clusters of words using a fuzzy clustering approach—Gaussian Mixture Model (GMM) with 200 clusters—the number of clusters is optimized via Bayesian Information Criterion which is known to be a sufficient estimate for GMM (Roeder and Wasserman, 1995). And finally, the posterior distribution of clusters given for the vector embedding of a word is used as features.

2.4.2. Topic Modeling

Topic modeling is a fuzzy clustering approach usually used to clusterize documents by topics. The very basic topic model—Probabilistic Latent Semantic Analysis (Hofmann, 1999) was employed. This model assumes that every document was drawn from a mixture of multinomial distributions over words. The components of the mixture are referred as topics. So, as a result of topic modeling, we obtain a distribution of words given the topic. Using Bayes' theorem we can easily compute the distribution of topics given the words. Finally, this distribution is used as a feature. The model was trained using a large unlabelled dataset of user's reviews. The tm[2] implementation was used.

2.4.3. Automatic Term Recognition

Since aspects are usually expressed by domain-specific terms, we check if the particular word-candidate is a part of domain-specific term. To do so, we apply methods for Automatic Term Recognition. Most of them, including those used by us, work as follows: take domain-specific text collection as an input; extract term candidates (n-grams filtered by the pre-specified part of speech patterns); compute features (e.g. frequency of term occurrences or tf-idf); and finally, classify or rank term candidates based on their feature vectors. In this work we skip the last step, i.e. we obtain the feature vector for each term candidate and then use it as follows: during a review text processing, we greedily search term candidates among word token sequences so that the longest appropriate term candidate is chosen, then we attach the corresponding feature vector to each word token from the matched sequence.

In particular, as an input text collection we use a combination of train and test data sets and also a set of documents crawled from the Web—namely, 44567 docs (82.6 Mb) from restoclub.ru for Restaurant domain and 7590 reviews (28.5 Mb) from otzovik.com for Automobile domain.

The following features are taken: 3 well-known features: Frequency; TF-IDF; C-Value (Frantzi et al., 2000) in modification that supports single-word terms (Lossio-Ventura et al., 2013); and 4 our features (Astrakhantsev, 2014): ExistsInKB—a boolean feature indicating if a term candidate is presented in Wikipedia; Link Probability—a probability of term candidate to be a hyperlink in Wikipedia; Key concept relatedness—a semantic relatedness value computed over Wikipedia to automatically found key concepts; PUATR—result of probabilistic Positive-Unlabeled classifier trained on top 100 term candidates (found by special method based on frequencies of nested occurrences) as positives and other candidates as unlabeled with all previously described features.

[2] https://github.com/ispras/tm

3. Evaluation

3.1. SVM parameter estimation

For SVM parameter estimation we perform 10-fold cross-validation on available training data with C parameter from 0.001 to 0.2 with step 0.001 in two settings (see Fig. 1). First settings is testing on training data (red line), the second settings is normal cross-validation (green line). As one can see, when to C < 0.045 F1 score grow for both train and test data.

For C > 0.45 F1 measure for train is grow and for test data it is stay almost same, thus we decided that this is frontier between over and underfitting. Thus we set C equals to 0.45

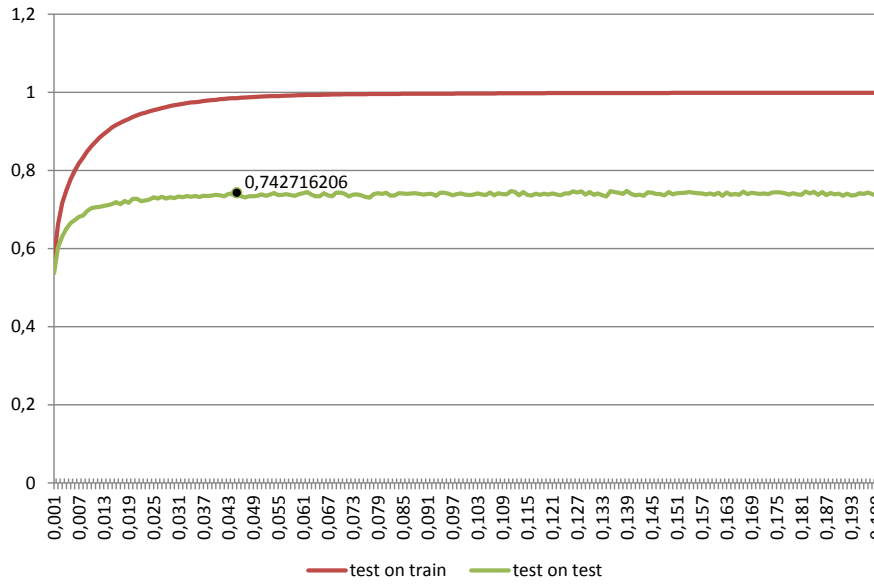


Fig 1. Method performance with different SVM parameter

3.2. Evaluation of feature groups impact

In order to understand impact of each feature group we sequentially remove each group from our feature set and measure method quality for task A. For quality measurement we perform repeated 10 times 10-fold cross-validation and compute 95% confidence interval for each quality metric. Results for automobile domain is presented in Table 1. Table 2 presents results for restaurant domain.

Table 1. Quality results (95% confidence intervals) for different features sets for Automobile domain (Task A)

	exact matching			partial matching		
features set	**precision**	**recall**	**f1**	**precision**	**recall**	**f1**
all	(0.7061; 0.7197)	(0.6500; 0.6618)	(0.6773; 0.6885)	(0.8080; 0.8200)	(0.6975; 0.7114)	(0.7493; 0.7604)
all—GloVe	(0.7107; 0.7249)	(0.6467; 0.6584)	(0.6775; 0.6891)	(0.8139; 0.8257)	(0.6888; 0.7015)	(0.7467; 0.7573)
all—TM	(0,7031; 0,7166)	(0,6427; 0,6548)	(0,6720; 0,6832)	(0,8061; 0,8181)	(0,6882; 0,7016)	(0,7431; 0,7540)
all—ATR	(0,7032; 0,7165)	(0,6414; 0,6537)	(0,6713; 0,6826)	(0,8066; 0,8185)	(0,6915; 0,7059)	(0,7452; 0,7565)
all—global	(0,7046; 0,7185)	(0,6509; 0,6633)	(0,6771; 0,6888)	(0,8068; 0,8190)	(0,6990; 0,7129)	(0,7496; 0,7609)
all—syntactic	(0,7132; 0,7276)	(0,6582; 0,6706)	(0,6850; 0,6968)	(0,8155; 0,8268)	(0,7069; 0,7203)	(0,7579; 0,7685)
all—NERC	(0,6373; 0,6535)	(0,5120; 0,5253)	(0,5682; 0,5810)	(0,7655; 0,7798)	(0,5812; 0,5968)	(0,6611; 0,6747)

Table 2. Quality results (95% confidence intervals) for different features sets for Restaurant domain (Task A)

	exact matching			partial matching		
features set	**precision**	**recall**	**f1**	**precision**	**recall**	**f1**
all	(0,7122; 0,7260)	(0,6546; 0,6692)	(0,6830; 0,6942)	(0,7894; 0,8024)	(0,7012; 0,7143)	(0,7439; 0,7530)
all—GloVe	(0,7146; 0,7284)	(0,6529; 0,6672)	(0,6831; 0,6943)	(0,7956; 0,8080)	(0,6963; 0,7093)	(0,7438; 0,7528)
all—TM	(0,7140; 0,7281)	(0,6450; 0,6591)	(0,6786; 0,6896)	(0,7912; 0,8045)	(0,6884; 0,7017)	(0,7375; 0,7467)
all—ATR	(0,7106; 0,7247)	(0,6514; 0,6662)	(0,6805; 0,6920)	(0,7887; 0,8020)	(0,6972; 0,7106)	(0,7414; 0,7507)
all—global	(0,7118; 0,7256)	(0,6551; 0,6696)	(0,6831; 0,6941)	(0,7893; 0,8017)	(0,7045; 0,7177)	(0,7458; 0,7545)
all—syntactic	(0,7101; 0,7249)	(0,6570; 0,6713)	(0,6833; 0,6949)	(0,7947; 0,8076)	(0,7009; 0,7144)	(0,7461; 0,7554)
all—nerc	(0,6325; 0,6488)	(0,5109; 0,5265)	(0,5656; 0,5795)	(0,7426; 0,7571)	(0,5775; 0,5929)	(0,6504; 0,6627)

As one can see, only NERC features make a meaningful contribution to the method. Other feature groups are not so significant.

3.3. Method performance on SentiRuEval testing dataset

The quality of proposed method trained on all available training data with all described feature groups are presented in table 3 for task A and in table 4 for Task B. These results are obtained by SentiRuEval organizers.

Table 3. SentiRuEval Task A experiment results

	exact matching			partial matching		
Domain	**precision**	**recall**	**f1**	**precision**	**recall**	**f1**
Automobile	0.760041	0.621793	0.676118	0.856055	0.655098	0.730366
Restaurant	0.723656	0.573800	0.631871	0.807759	0.616549	0.689096

Table 4. SentiRuEval Task B experiment results

	exact matching			partial matching		
Domain	**precision**	**recall**	**f1**	**precision**	**recall**	**f1**
Automobile	0.770100	0.553546	0.636623	0.866178	0.549210	0.659989
Restaurant	0.733599	0.513197	0.596179	0.814496	0.479988	0.590601

Conclusion

We have described aspect term extraction system, which employs SVM with a broad set of features. This system perform with high precision and good F1-measure on all settings and showed one of the best results among 21 runs received for aspect extraction tasks of SentiRuEval.

In addition, we made evaluation of impact of different feature groups and found that features used for named entity recognition are most useful for aspect extraction too. We also found that removing some features could slightly improve results of cross-validation. One of the reasons for such phenomena is sparsity of feature set. Therefore we can guess that feature selection and dimensionality reduction could improve quality of the proposed method. In addition, we should note that due to lack of time, we estimated SVM parameter only on full feature set and use it for all experiments. However SVM parameter estimation for each feature combination can improve overall performance of the system. This make a slot for future improvement of the proposed method.

References

1. *Astrakhantsev N.,* (2014), Automatic term acquisition from domain- specific text collection by using Wikipedia, The Proceedings of ISP RAS [Trudy ISP RAN], vol. 26, issue 4, P. 7–20.
2. *Fangtao L., Huang M., Zhu X.,* (2010), Sentiment analysis with global topics and local dependency, in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010).
3. *Frantzi K., Ananiadou S., Mima H.,* (2000), Automatic recognition of multi-word terms:. the c-value/nc-value method, International Journal on Digital Libraries, 3(2), 115–130.
4. *Jin Wei, Hung Hay Ho,* (2009), A novel lexicalized HMM-based learning framework for web opinion mining, in Proceedings of International Conference on Machine Learning (ICML-2009).
5. *Hofmann T.,* (1999), Probabilistic latent semantic indexing, in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, (pp. 50–57). ACM.
6. *Liu B.,* (2012), Sentiment analysis and opinion mining, Synthesis Lectures on Human Language Technologies, *5*(1), 1–167.
7. *Lossio-Ventura J. A., Jonquet C., Roche M., Teisseire M.,* (2013), Combining c-value and keyword extraction methods for biomedical terms extraction, In LBM'2013: 5th International Symposium on Languages in Biology and Medicine (pp. 45–49).
8. *Mei Qiaozhu, Xu Ling, Matthew Wondra, Hang Su, ChengXiang Zhai,* (2007), Topic sentiment mixture: modeling facets and opinions in weblogs, in Proceedings of International Conference on World Wide Web (WWW-2007).
9. *Niklas J., Gurevych I.,* (2010), Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields, in Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010).
10. *Nivre J., Hall J., Nilsson J., Chanev A., Eryigit G., Kübler S., Marsi E.,* (2007), MaltParser: A language-independent system for data-driven dependency parsing, Natural Language Engineering, *13*(02), 95–135.
11. *Pennington J., Socher R., Manning C. D.,* (2014), Glove: Global vectors for word representation, Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), *12*.
12. *Poria S., Cambria E., Ku L. W., Gui C., Gelbukh A.,* (2014), A rule-based approach to aspect extraction from product reviews, SocialNLP 2014, 28.
13. *Popescu A. M., Etzioni O.,* (2007), Extracting product features and opinions from reviews, In Natural language processing and text mining (pp. 9–28), Springer London.
14. *Qiu G., Liu B., Bu J., Chen C.,* (2011), Opinion word expansion and target extraction through double propagation, Computational linguistics, 37(1), 9–27.
15. *Ratinov L., Roth D.,* (2009) Design challenges and misconceptions in named entity recognition, Proceedings of the Thirteenth Conference on Computational Natural Language Learning / Association for Computational Linguistics, pp. 147–155.

16. *Roeder K., Wasserman L.,* (1997), Practical Bayesian density estimation using mixtures of normals, Journal of the American Statistical Association, 92(439), 894–902.
17. *Scaffidi C., Bierhoff K., Chang E., Felker M., Ng H., Jin C.,* (2007), Red Opal: product-feature scoring from reviews, In Proceedings of the 8th ACM conference on Electronic commerce, pp. 182–191
18. *Turdakov D., Astrakhantsev N., Nedumov Y., Sysoev A., Andrianov I., Mayorov V., Fedorenko D., Korshunov A., Kuznetsov S.* (2014), Texterra: A Framework for Text Analysis, Proceedings of the Institute for System Programming of RAS [Trudy ISP RAN], volume 26, Issue 1, pp. 421–438.
19. *Yejin C., Cardie C.,* (2010), Hierarchical sequential learning for extracting opinions and their attributes, in Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010).
20. *Zhang T., Johnson D.* (2003), A robust risk minimization based named entity recognition system, Proceedings of the seventh conference on Natural language learning at HLTNAACL 2003-Volume 4 / Association for Computational Linguistics, pp. 204–207.

АВТОМАТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ ТОНАЛЬНОСТИ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ СЕМАНТИЧЕСКИХ ШАБЛОНОВ И СЛОВАРЕЙ ТОНАЛЬНОЙ ЛЕКСИКИ

Поляков П. Ю. (pavel@rco.ru),
Калинина М. В. (kalinina_m@rco.ru),
Плешко В. В. (vp@rco.ru)
ООО «ЭР СИ О», Москва, Россия

Ключевые слова: определение тональности, анализ мнений, тональность объектов, тональность атрибутов, синтактико-семантический анализ, семантические шаблоны

AUTOMATIC OBJECT-ORIENTED SENTIMENT ANALYSIS BY MEANS OF SEMANTIC TEMPLATES AND SENTIMENT LEXICON DICTIONARIES

Polyakov P. Yu. (pavel@rco.ru),
Kalinina M. V. (kalinina_m@rco.ru),
Pleshko V. V. (vp@rco.ru)
RCO LLC, Moscow, Russia

This paper studies use of a linguistics-based approach to automatic object-oriented sentiment analyses. The original task was to extract users' opinions (positive, negative, neutral) about telecom companies, expressed in tweets and news. We excluded news from the dataset because we believe that formal texts significantly differ from informal ones in structure and vocabulary and therefore demand a different approach. We confined ourselves to the linguistic approach based on syntactic and semantic analysis. In this approach a sentiment-bearing word or expression is linked to its target object at either of two stages, which perform successively. The first stage includes usage of semantic templates matching the dependence tree, and the second stage involves heuristics for linking sentiment expressions and their target objects when syntactic relations between them do not exist. No machine learning was used. The method showed a very high quality, which roughly coincides with the best results of machine learning methods and hybrid approaches (which combine machine learning with elements of syntactic analysis).

Key words: sentiment analysis, object-oriented sentiment analysis, aspect-based sentiment analysis, opinion mining, syntactic and semantic analysis, semantic templates

1. Introduction

The task of automatic sentiment analysis of natural language texts has become extremely in demand. Many commercial companies producing goods and services are interested in monitoring social networking websites and blogs for users' opinions about their products and services. However, until recently there were no tagged text corpora in Russian on which developers could test and compare quality of their methods. This gap was filled by ROMIP and later SentiRuEval sentiment analysis evaluation conferences with their sentiment analysis tracks. However, the task of the previous conferences was to detect general sentiment of a text (for example, see Chetviorkin I., Braslavski P. I., Loukachevitch N. [2]), while at the present conference the task was brand new—object-oriented sentiment analysis, which is more difficult and requires more sophisticated algorithms; for, in case of general sentiment detection, selection of positive and negative terms and defining of their weights are important, while, in case of object-oriented sentiment detection, syntactic relations between a target object and a word expressing sentiment are also of great importance.

Such object-oriented method is not new for us; we have already used similar approach in our previous research. For instance, we evaluated sentiment-oriented opinions in regard to car makes on the material of the LiveJournal blog AUTO_RU (see description of the method in Ermakov A. E. [4]). It should be mentioned, however, that in all the previous cases results had only been evaluated by ourselves. Participation in SentiRuEval gave us a chance to have an independent evaluation of our method and compare our results with other participants'.

In this paper we present results of applying a linguistics-based approach involving syntactic and semantic analysis to the task of automatic object-oriented sentiment analysis. We confined ourselves to a linguistic method only, having excluded machine learning, because it was interesting to see what results a pure linguistic approach without machine learning methods would provide.

The task was to find sentiment-oriented opinions (positive and negative) about telecom companies in tweets.

2. Related Work

Usually object-oriented or aspect-oriented approaches either rely only on statistics-based algorithms, word distance count, machine learning, etc. to find opinion targets (starting with the first work on opinion target extraction by Hu and Liu [5]); or they may use shallow parsing to segment a sentence, find significant conjunctions, negations, and modifiers (ex., Kan D. [7]). Other approaches are looking for syntactic dependency between a sentiment term and its target (ex., Popescu A., Etzioni O. [9]), ignoring sentiment-bearing words which are not syntactically related to any target object. The distinctive feature of our approach is that using a deep linguistic method we take into account not only syntactically related sentiment terms (which provides high precision) but also independent sentiment-bearing words and phrases (which provides high recall).

Some researchers try combine statistical and linguistic methods in order to achieve the best results; for example, in Jakob N., Gurevych I. [6] authors use, among other, the dependency parse tree to link opinion expressions and the corresponding targets; and the experiments show that adding the dependency path based feature yields significant improvement to their method. However, their algorithm is searching for short and direct dependency relations only; therefore, their approach has difficulties with more complex sentences. Furthermore, they do not distinguish between a target object (ex., *camera*), its attributes or parts (ex., *lens cap, strap*), and its qualities (ex., *usability*); and, hence, they label the closest noun phrase as a target of the opinion. In contrast, we use a very basic ontology to distinguish between a target object, attributes, and qualities; and having found a sentiment related to an attribute or quality our algorithm goes down the dependency parse tree searching for a target object. If not found syntactically, the target object is being searched for by a heuristic, based on the clause distance. When the target object is found, the sentiment labeled to its attribute is assigned to the object.

3. Methods

To perform the task we based on our previous researches and solutions. Detailed description of these methods can be found in Ermakov A. E., Pleshko V. V. [3] and Ermakov A. E. [4]. New to the approaches described in [3] and [4] was adding so-called 'Free Sentiment Detection', which will be described in Section 3.2.

The text analysis algorithm has the following stages in regard to the sentiment detection task:

1) Tokenization;
2) Morphological analysis;
3) Object extraction;
4) Syntactic analysis;
5) Fact extraction (use of semantic templates);
6) Free sentiment detection.

Stages 1, 2, and 4 were implemented by standard RCO tools for general text analysis. At stage 3 we paid more attention to the objects concerning the given subject (names of mobile companies, telecom terminology, etc.). Stages 5 and 6 were core to the sentiment detection task and, therefore, will be described in detail.

3.1. Semantic Templates

The main method of sentiment analysis involved usage of semantic templates.

Semantic template is a directed graph representing a fragment of a syntactic tree with certain restrictions applied to its nodes. The syntactic tree of a sentence contains semantic and syntactic relations between words, which are defined by the syntactic parser. The restrictions in the templates can be applied to a part of speech, name, semantic type, syntactic relations, morphological forms, etc. Fact extraction is performed by finding a subgraph in the syntactic tree of a sentence which is isomorphic to the template (with all restrictions applied).

RCO syntactic analyzer, based on the dependency tree approach, has been used. The semantic network built by the syntactic parser is invariant to the word order and voice; for example, sentences (1) *Оператор украл деньги со счета* and (2) *Деньги украдены оператором со счета* will have the same semantic net. Such semantic network constitutes an intermediate representation level between the semantic scheme of a situation and its verbal expression, that is, a deep-syntactic representation, abstracted from the surface syntax.

Settings of the semantic interpreter allow filtering negative and 'unreal' (imperative, conditional, etc.) statements, which don't correspond to real events and should not be analyzed. As a result, examples like (3) *если Билайн будет плохо работать; сеть якобы падает; связь бы обрывалась; не Билайн плохо работает* can be excluded from the sentiment detection.

To decrease the number of templates describing semantic frames, we have so-called auxiliary templates, which add new nodes and relations into the semantic network. In the process of semantic analysis and fact extraction auxiliary templates work before all other templates, so that semantic templates can base on the net built by both the syntactic analyzer and the auxiliary templates. For example, if we interpret phrases like (4) *X does Y, X begins to do Y,* and (5) *X decides to do Y* as equal for a particular semantic frame, instead of creating a semantic template for each example we can have one auxiliary template, which will mark the subject of the main verb as the subject of the subordinate verb, and one simple semantic template—(4) *X does Y.*

Semantic templates can have so-called 'forbidding nodes' which impose restrictions on the context, defining in which context the template should not match. For example, (6) *У Билайна надежная связь* is a positive statement, while adding the adverb *наименее* changes its sentiment to opposite: (7) *У Билайна наименее надежная связь*. By the means of forbidding nodes we can distinguish between these two sentences, stating that the adjective should not be modified by the adverb *наименее*. Usage of forbidding nodes significantly increases the precision of sentiment analysis.

Fig. 1 demonstrates a semantic template used to detect sentiment expressed by a verb or adverb in sentences like: (8) *Билайн ловит хорошо; Интернет летает*.

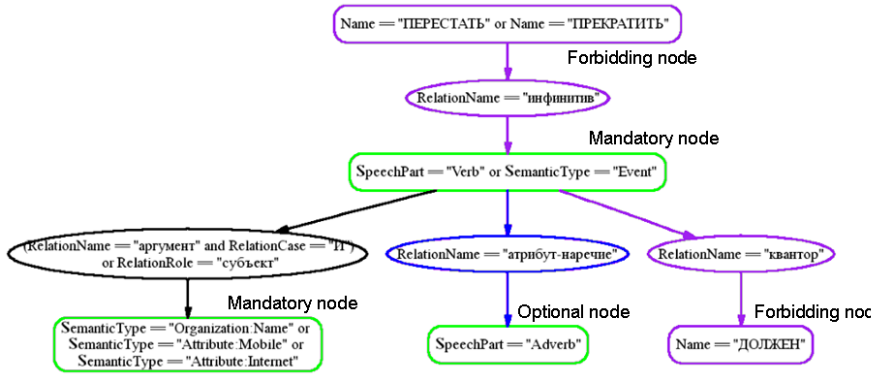


Fig. 1. Example of a semantic template

Nodes contain restrictions on parts of speech (*SpeechPart == "Verb"; SpeechPart == "Adverb"*), lexical items (*Name == "ПЕРЕСТАТЬ" or Name == "ПРЕКРАТИТЬ"*), semantic categories (*SemanticType == "Organization:Name" or SemanticType == "Attribute:Mobile"*). Restrictions on semantic and syntactic relations between words include: relation name (*RelationName == "аргумент»; RelationName == «квантор»*), semantic role (*RelationRole == "субъект»*), case (*RelationCase == "И"*). Forbidding nodes state that the verb expressing sentiment should not be controlled by the verbs *перестать* or *прекратить* or modified by the predicative *должен*. Thus, this template will match the sentence (8) *Билайн хорошо ловит* (which is positive), but not (9) *Билайн перестал хорошо ловить* (which is negative) or (10) *Билайн должен хорошо ловить* (which we consider neutral).

Restrictions of the semantic templates were enriched by the use of special dictionaries (so-called filters), containing vocabulary for positive and negative appraisals. This vocabulary includes nouns, adjectives, verbs, adverbs, and collocations. A word from a filter must be syntactically related to the target of evaluation. Selection of terms for the filters was manual, performed by a linguistic expert. Examples of positive terms: *супербыстрый, шустро, красота, крутяк, блистать, радовать, обеспечивать уверенный прием*. Examples of negative terms: *завышенный, препротивнейший, позорище, тормознутость, обдирать, терять соединение, фигово.*

For example, a set of particular words from the semantic filters are applied to the template in Fig.1 as restrictions: verbs or verbal nouns parameterize the node with the restriction *SpeechPart == "Verb" or SemanticType == "Event"*; adverbs parameterize the node with the restriction *SpeechPart == "Adverb"*, both these nodes have the semantic role 'Appraisal'.

Ultimate targets of evaluation were main Russian mobile phone providers (Beeline, Megafon, MTS, Rostelecom, Tele2), but also users' appraisals of providers' attributes were taken into account (communication quality, mobile Internet, customer service, etc.).

Analyzing users' comments and opinions on social networking sites and forums experts defined a set of attributes which were most frequently mentioned by mobile phone users. Thus, a list of most important things for users was made. Given attributes were divided into three classes: 1) Mobile Attributes—terms strictly connected to the mobile telephony: *SMS, MMS, 3G, LTE, SIM-card, roaming, etc.*; 2) Internet Attributes—terms strictly connected to the Internet: *Internet, ping, etc.*; 3) General Attributes—terms often used related to the mobile telephony but which can also refer to other domains: *call center, signal, network, customer support, balance, etc*. Each list was extended by synonyms and spelling variants (*интернет=инет=и-нет; lte=лте =ltешечка =лте-шечка; баланс счета=состояние счета=средства на счету=деньги на счету, etc.*). When a sentiment related to a certain attribute was detected, given sentiment was also ascribed to the corresponding mobile provider.

In Fig.1 the node with the restriction *SemanticType == "Organization:Name" or SemanticType == "Attribute:Mobile" or SemanticType == "Attribute:Internet"* is parameterized by names of mobile operators, mobile attributes or Internet attributes; the semantic role of the node is 'Target Of Evaluation'.

This method provides a very high precision, though not so high recall.

3.2. 'Free' Sentiment

Although usage of semantic templates provides very good accuracy, this method has its disadvantage—a word expressing sentiment must be in the same sentence as the target of evaluation and must be syntactically related to it. As it is not always so in natural texts, some cases of clearly expressed sentiment will be omitted by this method, and the recall will suffer. This problem becomes extremely significant when we analyze informal texts—forums, social networking websites, blogs, etc. Writing an informal text message, users often disregard punctuation and spelling rules, mistype, because of which the syntactic parser may fail to correctly analyze the structure of a sentence and build a semantic network. Users often express their sentiment through interjections, which are not a part of the syntactic tree; hence the semantic templates are of no use in this case. We call words that express sentiment but have no syntactic relation to the target of evaluation (or such relation has not been built by the parser) 'free sentiment'.

To solve this problem another method has been applied. We used an algorithm which is looking for free sentiment in the text using dictionaries (or profiles) of positive and negative lexicon, and if such sentiment has been found tries to relate it to the target object.

These two methods complement each other, with the semantic template method working first. In this regard, the classifier 'ignores' terms already found and related to the target object by templates, because we assume that the accuracy provided by the semantic templates is close to 100%.

As profiles for positive and negative classes we used corresponding filters, having removed context-dependent sentiment words and leaving only explicit emotional or evaluative vocabulary. For example, we removed verbs *УМЕРЕТЬ, ПРОИГРЫВАТЬ*, because although they are obviously negative in the context like: (11) *интернет умер;* (12) *оператор X проигрывает оператору Y*; but in another context, not related to the mobile telephony, they may be neutral and just state a fact. At the same time we enriched our profiles with interjections and other emotional expressions which cannot be syntactically related to the object of evaluation, for example: (13) *не надо так! что за нах; ни фига себе; ну как так можно, etc*.

Having found a sentiment, our algorithm was looking for an object of evaluation—a name of a mobile company—in the given text and ascribed this sentiment to the target. If several mobile operators were mentioned in the text, the appraisal was ascribed to the nearest operator. If both positive and negative sentiment was detected related to the same mobile provider mentioned, we gave preference to the negative sentiment, regarding positive expressions as sarcasm.

No machine learning had been used. The methods applied were based on linguistic analysis only.

4. Dataset

The training and test collection granted by organizers consisted of 5,000 labeled and 5,000 not labeled tweets containing sentiment-oriented opinions or positive and negative facts about telecom companies.

As the main goal of social networks sentiment analysis is to find sentiment-oriented opinions, we labeled texts containing reprints of news and additionally measured sentiment detection quality for the training collection with news reprints excluded. We excluded news texts from the final dataset because we believe that the difference in structure and vocabulary between formal (news) and informal (posts, blogs, tweets) texts is crucial. As a rule, in news texts authors don't express their attitude openly; news is more likely to contain coverage of events and facts, which can be interpreted as positive or negative for the newsmaker, rather than explicit sentiment; and therefore analyzing news demand a different approach. Furthermore, vocabulary of informal texts is quite different from vocabulary of formal texts.

That is why we additionally estimated the method performance on the collection with news reprints and companies' press releases excluded from the dataset. Since our method is based on linguistic analysis only, we did not use training collection.

5. Results

Initially, for the purpose of estimation of coincidence between assessors we asked our expert to evaluate the test collection manually and marked each reference to mobile phone companies as being positive, negative or neutral. Results of our expert's evaluation are presented in Table 1. F1-measure macro- and micro-averaged was used as a primary evaluation metric [1]. Additionally, for convenience, recall and precision are also present in the tables. As shown in Table 1, the estimation of tweets by our expert differed from one granted by the organizers. We consider the score given by our expert as the highest possible for an automatic sentiment detection system for the given collection. The agreement between our expert and organizers' labeling was higher when we excluded news from the dataset, which confirms our assumption that a different approach should be used for sentiment analysis of news.

Table 1. The estimation of coincidence between expert and assessors

	Macro-average			**Micro-average**		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**
With news	0.722	0.686	0.703	0.771	0.728	0.749
Without news	0.785	0.694	0.737	0.831	0.735	0.780

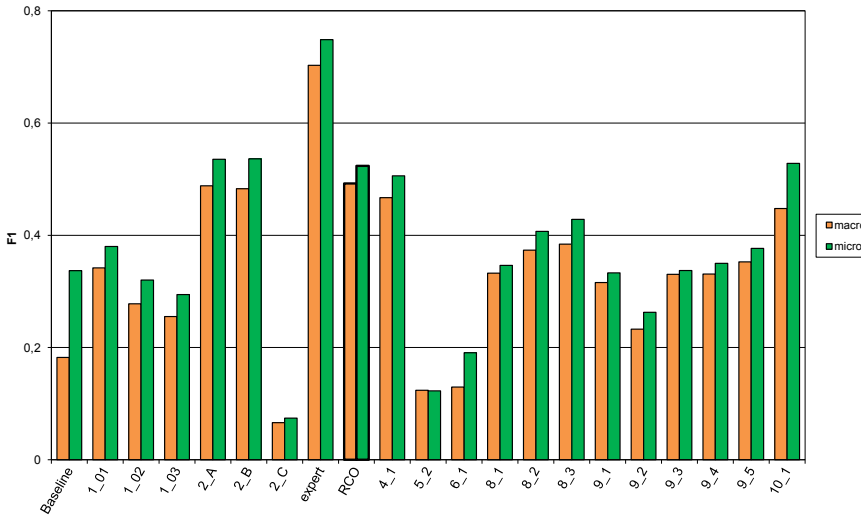


Fig. 2. Macro- and micro-averaged F1 measure calculated on test collection for all participants. The scores for our method are labeled as "RCO". The scores of expert's evaluation are labeled as "expert"

The results of all participants are shown in Fig. 2, our results are highlighted by bold lines and are labeled as "RCO". It is interesting that several methods probably based on different approaches demonstrate very similar high scores of F1 (about 0.5), nevertheless, these scores are sufficiently less than theoretical maximum that corresponds to coincidence between assessors (see bars "Expert" on Fig. 2). It could prove that automatic sentiment detection task is still a challenging problem.

The detailed results of our method are presented in Table 2. We calculated recall, precision and F1 for original collection (labeled as "With news") and for collection with exclusion of messages contained news and press releases (labeled as "Without news"). For comparison, the best scores among the methods of all participants are presented.

Table 2. The performance of our method and best F1 measure among the methods of all participants

	Macro-average			Micro-average		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**
With news	0.436	0.566	0.480	0.451	0.585	0.509
Without news	0.465	0.562	0.492	0.475	0.583	0.524
Best result			0.492			0.536

6. Conclusion

Our combined linguistic method showed a very high quality, which roughly coincides with the best results of machine learning methods and hybrid approaches (combining machine learning with elements of syntactic analysis). In the future we are planning to add machine learning to our linguistic approach.

References

1. *Blinov P. D., Kotelnikov E. V.* (2014), Using distributed representations for aspect-based sentiment analysis, Dialog '14, Bekasovo.
2. *Chetviorkin I., Braslavski P. I., Loukachevitch N.* (2012), Sentiment analysis track at ROMIP 2011, Bekasovo.
3. *Ermakov A. E., Pleshko V. V.* (2009), Abstract Semantic Interpretation in Computer Text Analysis Systems [Semanticheskaya interpretatsiya v sistemakh kompyuternogo analiza teksta], Information Technologies [Informacionnye tehnologii], Vol. 6, pp. 2–7.
4. *Ermakov A. E.* (2009), Knowledge Extraction from Text and its Processing: Current State and Prospects [Izvlecheniye znaniy iz teksta i ikh obrabotka: sostoyaniye i perspektivy], Information Technologies [Informacionnye tehnologii], Vol. 7, pp. 50–55.
5. *Hu M., Liu B.* (2004), Mining and summarizing customer reviews, International Conference on Knowledge Discovery and Data Mining (ICDM).
6. *Jakob N., Gurevych I.* (2010), Extracting Opinion Targets in a Single-and Cross-Domain Setting with Conditional Random Fields, Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010).
7. *Kan D.* (2012), Rule-based approach to sentiment analysis at ROMIP '11 , Bekasovo.
8. *Loukachevitch N., Blinov P., Kotelnikov E., Rubtsova Yu., Ivanov V., Tutubalina E.* (2015), SentiRuEval Testing Object-Oriented Sentiment Analysis Systems in Russian.
9. *Popescu A., Etzioni O.* (2005), Extracting product features and opinions from reviews, Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP) .
10. *Polyakov P. Yu., Kalinina M. V., Pleshko V. V.* (2012), Research of applicability of thematic classification to the problem of book review classification. Dialog '12. Naro-Fominsk.
11. *Polyakov P. Yu., Frolov A. V., Pleshko V. V.* (2013), Using semantic categories in application to book reviews sentiment analysis, Dialog'13, Bekasovo.

ГЛУБОКИЕ РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ АСПЕКТНО-ОРИЕНТИРОВАННОГО АНАЛИЗА ТОНАЛЬНОСТИ ОТЗЫВОВ ПОЛЬЗОВАТЕЛЕЙ НА РАЗЛИЧНЫХ ЯЗЫКАХ

Тарасов Д. С. (dtarasov3@gmail.com)

Интернет-портал reviewdot.ru, Казань, Россия

Ключевые слова: рекуррентные нейронные сети, анализ тональности, извлечение аспектных терминов, унифицированный подход

DEEP RECURRENT NEURAL NETWORKS FOR MULTIPLE LANGUAGE ASPECT-BASED SENTIMENT ANALYSIS OF USER REVIEWS

Tarasov D. S. (dtarasov3@gmail.com)

Reviewdot research,Kazan,Russian Federation

Deep Recurrent Neural Networks (RNNs) are powerful sequence models applicable to modeling natural language. In this work we study applicability of different RNN architectures including uni- and bi-directional Elman and Long Short-Term Memory (LSTM) models to aspect-based sentiment analysis that includes aspect terms extraction and aspect term sentiment polarity prediction tasks. We show that single RNN architecture without manual feature-engineering can be trained to do all these subtasks on English and Russian datasets. For aspect-term extraction subtask our system outperforms strong Conditional Random Fields (CRF) baselines and obtains state-of-the-art performance on Russian dataset. For aspect terms polarity prediction our results are below top-performing systems but still good for many practical applications.

Keywords: recurrent neural networks, sentiment polarity, aspect term extraction, unified approach

1. Introduction

In many practical natural language processing (NLP) systems, it is desirable to have one architecture that can be quickly adapted to different tasks and languages without the need to design new feature sets. Recent success of deep neural networks in general and deep RNNs in particular offers hope that this goal is now within reach. RNNs were applied to a number of English NLP problems, demonstrating their superior capabilities in slot-filling task [Mesnil et al, 2013] and opinion mining [Irsoy and Cardie, 2014].

While these results are promising it is still unclear if RNNs can now be used to replace other models in practical multi-purpose NLP system and if single RNN architecture can efficiently perform many different tasks.

Our work evaluates a number of RNN architectures on three different datasets: ABSA Restaurants (English) dataset from SemEval-2014 [Pontiki et al, 2014] and two Russian datasets (Restaurants and Cars) from SentiRuEval-2015.

We show that RNN performance on aspect terms extraction is close to state-of-the art and results on sentiment prediction, while being significantly behind top performing systems, outperform strong baselines and offer sufficient performance for use in practical applications. We discuss factors that contribute to RNNs results and suggest possible directions to further improve their performance on these tasks.

2. Related work

Sentiment analysis or opinion mining is the computational study of people's attitudes toward entities. In user reviews analysis two principal tasks are aspect terms extraction and aspect sentiment polarity prediction.

Aspect term extraction methods could roughly be divided into supervised and unsupervised approaches. In supervised approach aspect extraction is usually seen as sequence labeling problem, and often solved using variants of conditional random field (CRF) [Ganu et al, 2009;Breck and Cardie, 2007] methods, including semi-CRF systems, that operate at the phrase level and thus allow incorporation of phrase-level features [Choi and Cardie, 2010]. Such systems currently hold state-of-the arts results in term extraction from user reviews [Pontiki et al, 2014]. However, success of CRF and semi-CRF approaches depends on the access to rich feature sets such as dependency parse trees, named-entity taggers and other preprocessing components, that are often not readily available in underresourced languages such as Russian. Unsupervised approaches to term extraction attempts to cut cost and effort associated with manual feature selection and annotation of training data. These approaches typically utilize topic models such as Latent Dirichlet Allocation to learn aspect terms [Brody and Elhadad, 2010]. Their performance however, is below that of supervised systems trained on in-domain data.

Quite recently recurrent neural network models were proposed to solve sequence tagging problems, including similar opinion mining task [Irsoy and Cardie, 2014], demonstrating results superior to all previous systems. Importantly, these results were obtained using only word vectors as features, eliminating the need for complex feature-engineering schemes.

Similarly, sentiment polarity prediction subtask is solved within supervised and unsupervised learning frameworks. State-of-the-art performance on term polarity detection is currently obtained by using support vector machines (SVM) with rich feature sets that include parse trees and large opinion lexicons, together with preprocessing to resolve negation [Pontiki et al, 2014]. Unsupervised methods in sentiment analysis usually focus on construction of polarity lexicons for which number of approaches currently exists [Brody and Elhadad, 2010], and then applying heuristics to determine term polarity.

Neural network based methods were developed recently to detect document level and phrase-level sentiment, including tree-based autoencoders [Socher et al, 2011;2013] and convolutional neural networks [dos Santos and Gatti, 2014;Blunsom et al, 2014] and Elman-type RNNs were applied to sentence-level sentiment analysis with promising results [Wenge et al, 2014].

3. Methodology

3.1. Datasets

SemEval-2014 ABSA Restaurants dataset [Pontiki et al, 2014] was downloaded through MetaShare (http://metashare.ilsp.gr:8080/). This dataset is a subset of (Ganu et al, 2009) dataset. It contains English statements from restaurants reviews (3,041 in training and 800 sentences in test set) annotated for aspect terms occurring in the sentences, aspect term polarities, and aspect category polarities.

Russian Restaurants dataset and corresponding Cars dataset released by SentiRuEval-2015 organizers to participants consist of similarly annotated reviews in Russian with a number of important differences. These datasets contain whole reviews, rather than individual sentences and are annotated with three categories of aspect terms “explicit” (roughly equivalent to SemEval-2014 notion of aspect term), “implicit” and so called “polarity facts”—statements that don't contain explicit judgments but nevertheless tell something good or bad about aspect in question.

Auxiliary dataset for training Russian unsupervised word vectors was constructed from concatenation of unannotated cars and restaurants reviews, provided by SentiRuEval-2015 organizers and 300,000 user reviews of various consumer products from reviewdot.ru database (obtained by crawling more than 200 online shops and catalogs).

3.2. Evaluation of human disagreement

As a part of this work we decided to evaluate human disagreement on SentiRuEval-2015 Restaurants dataset because we found many examples that seemed ambiguous. To do this we split dataset in two parts (70/30) and appointed two human judges. Human judges were given “annotation guidlines” sent by SentiRuEval organizers and 70% of annotated dataset. They then were asked to annotate remaining 30% with aspect terms (explicit, implicit and polar facts) and results were compared to original annotation using evaluation metrics described in “metrics” section.

3.3. Recurrent neural networks

A recurrent neural network [Elman, 1990] is a type of neural network that has recurrent connections. This makes them applicable for sequential prediction tasks, including NLP tasks. In this work, we consider simple Elman-type networks and Long-Short Term Memory architectures.

3.3.1. Simple recurrent neural network

In an Elman-type network (Fig. 1a), the hidden layer activations $h(t)$ at time step t are computed by transformation of the current input layer $x(t)$ and the previous hidden layer $h(t-1)$. Output $y(t)$ is computed from the hidden layer $h(t)$.

More formally, given a sequence of vectors {x (t)} where t = 1..T, an Elman-type RNN computes memory and output sequences:

$$h(t) = f(Wx(t) + Vh(t-1) + b) \tag{1}$$

$$y(t) = g(Uh(t) + c) \tag{2}$$

where f is a nonlinear function, such as the sigmoid or hyperbolic tangent function and g is the output function. W and V are weight matrices between the input and hidden layer, and between the hidden units. U is the output weight matrix, b and c are bias vectors connected to hidden and output units. $h(0)$ in equation (1) can be set to constant value that is chosen arbitrary or trained by backpropagation.

Deep RNN can be defined in many possible ways [Pascanu et al, 2013], but for the purposes of this work deep RNNs were obtained by stacking multiple recurrent layers on top of each other.

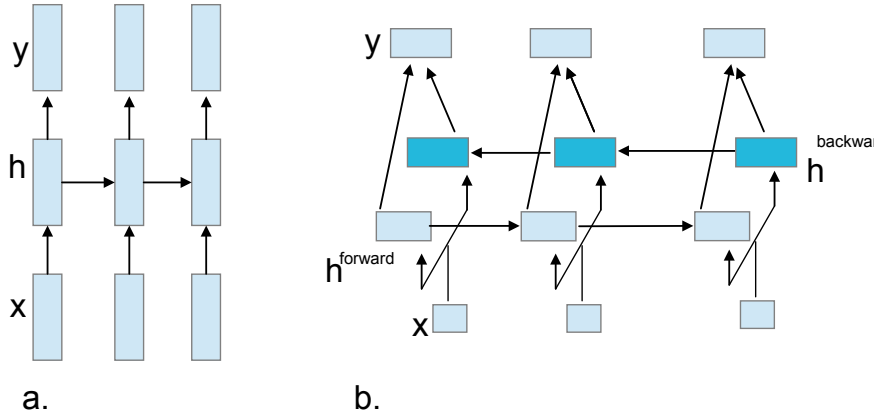


Figure 1. Recurrent neural networks, unfolded in time in three steps
a. Simple recurrent neural network b. Bidirectional recurrent neural network

3.3.2. Long Short Term Memory

The structure of the LSTM [Hochreiter and Schmidhuber, 1997] allows it to train on problems with long term dependencies. In LSTM simple activation function *f* from above is replaced with composite LSTM activation function. Each LSTM hidden unit is augmented with a state variable *s(t)* The hidden layer activations correspond to the 'memory cells' scaled by the activations of the 'output gates' *o* and computed in following way:

$$h(t) = o(t) * f(c(t)) \tag{3}$$

$$c(t) = d(t) * (c(t-1) + i(t)) * f(Wx(t) + Vh(t-1) + b) \tag{4}$$

where * denotes element-wise multiplication, *d (t)* is dynamic activation function that scales state by "forget gate" and *i (t)* is activation of input gate.

3.3.3. Bidirectional RNNs

In contrast with regular RNN that can only consider information from past states, bidirectional recurrent neural network (BRNN) [Schuster and Kuldip, 1997] can be trained using all available input data in the past and future. In BRNN (Fig. 1b) neuron states are split in a part responsible for positive time direction (forward states) and and a part for the negative time direction (backward states):

$$h(t)^{forward} = f(W^{forward} x(t) + V^{forward} h^{forward}(t-1) + b^{forward}) \tag{5}$$

$$h(t)^{backward} = f(W^{backward} x(t) + V^{backward} h^{backward}(t+1) + b^{backward}) \tag{6}$$

$$y(t) = g(U^{forward} h^{forward} + U^{backward} h^{backward} + c) \tag{7}$$

3.3.4. Training

All networks were trained using backpropagation through time (BPTT) [Werbos, 1990] algorithm with mini-batch gradient descent with one sentence per mini-batch as suggested in [Mesnil et al, 2013]. For sequence labeling tasks loss function was evaluated at every timestep, while for classification tasks such as term polarity prediction, loss function was only evaluated at the position corresponding to terms whose polarity was being predicted.

3.3.5. Regularization

To prevent overfitting small Gaussian noise was added to network inputs. Large networks were also regularized with dropout [Hinton et al, 2012] a recently proposed technique that omits certain proportion of the hidden units for each training sample.

3.4. Word embeddings

Real-valued embedding vectors for words were obtained by unsupervised training of Recurrent Neural Network Language Model (RNNLM) [Mikolov et al, 2010].

English embeddings of size 80 trained on 400M Google News dataset were downloaded from RNNToolkit (http://rnnlm.org/) website. Russian embeddings of same size were trained using auxiliary dataset described above, using same method. Russian text was preprocessed by replacing all numbers with #number token and all occurrences of rare words were replaced by corresponding word shapes.

3.5. Evaluation metrics

For term extraction tasks where term boundaries are hard to identify even for humans, it is generally recommended to use soft measures like Binary Overlap that counts every overlapping match between a predicted and true expression as correct [Breck et al, 2007], and Proportional Overlap that computes partial correctness proportional to the overlapping amount of each match [Johansson and Moschitti, 2010].

From the description of SemEval-2014 task it appears that exact version of F- measure was used (only exact matches count), even though organizers note that "In several cases, the annotators disagreed on the exact boundaries of multi-word aspect terms".

For Russian SentiRuEval-2015 datasets, due to somewhat different annotation approach, multi-word (4 and 5 word terms) are quite common and human disagreement is quite large (as will be shown below). SentiRuEval-2015 organizers adopt two metrics for aspect-term extraction—main (based on exact count) and secondary (based on proportional overlap).

In SentiRuEval-2015 datasets all terms are tagged as "relevant" (related to target entity), or irrelevant (related to something else) and official metrics only count identification of relevant terms as correct. We feel that identification of aspect term and classification it as "relevant" or not are two fundamentally different tasks and should be measured separately. Due to extremely low presence (less than 5%) of irrelevant terms, their exclusion is quite hard for machine learning algorithm to achieve, and finding algorithms that do that well is a problem of significant theoretical interest. Such systems cannot be identified using official metrics, since contribution of "relevance" detection to overall F1 value is rather small.

For the purposes of this paper unless otherwise stated, we apply F-measure based on proportional overlap to facilitate comparison of results obtained on different datasets. For English Restaurants ABSA dataset F-measure is computed on Test dataset of 800 sentences (that was not used in development of models). For Russian datasets, as test data were not available at the time of this work, we separate development set of 5000 words and use 7-fold cross-validation on remaining data, similar to [Isroy and Cardie, 2014] approach. Since we participated in a number of SentiRuEval-2015 tracks, official results according to SentiRuEval-2015 metrics are also shown for comparison and discussion purposes.

For classification tasks such as sentiment polarity and aspect category detection tasks, macro average of F-measure cannot be used due to the fact that some categories (such as "conflict" polarity, named "both" in Russian dataset) are extremely rare (Russian Restaurant dataset contains less than 80 instances of "both" polarity per 3,000 instances of aspect terms). F-measure for such categories is subject to huge sampling

error, and can also be undefined (with zero precession and recall), making macro average value undefined also. To prevent this problem from occurring SemEval-2014 uses Accuracy instead of F-measure. SentiRuEval-2015 organizers use F1 micro average in addition to macro average. In this paper, for classification tasks we show overall accuracy, computing macro-average as additional measure where possible.

3.6. Baselines

For term extraction task we consider several baseline systems: simple feed-forward multi-layer perceptron (MLP), frame-level MLP (a feed-forward MLP with inputs of only word embedding features within a word context window), logistic regression using word embedding features, and CRF using stemmed words and POS-tags as features.

4. Results and Discussion

4.1. Aspect term extraction task

Tables 1–3 summarize our results on aspect term extraction. Initially, for Russian Restaurant dataset, we found it very difficult to improve upon simple CRF baseline. Manual examination of annotation revealed a number of inconsistent decisions in provided training data, for example in one place term "официантка Любовь" ("servant Lubov") was tagged as a whole, while in other similar case servant name was not tagged as part of the term. That led us to evaluation of human disagreement that appeared to be very close to baseline results, making term extraction very formidable challenge.

Nevertheless, we found that augmented forward RNN outperforms CRF baseline on explicit aspect extraction and deep LSTM model outperforms both CRF and Frame-NN baselines on all subtasks, while simple BRNN while providing reasonable good results, failed to improve on these baselines in contrast with English dataset. We think that inconsistent annotation in training set leads to over-fitting in simple BRNNs, because complex local models are learned before long time dependencies in the data can be discovered.

Overall, as shown in Table 2, our system obtains best result in extraction of all aspects terms according to proportional measure and best result in extraction of all aspect terms on cars dataset according to exact measure, while holding second-best result on restaurants dataset. These good results, should, however, be interpreted with caution due to relatively small number of participants, general lack of strong competitors and poor quality of the data (at least in Restaurant domain).

Therefore, to better understand system capabilities we evaluated our system on English dataset of SemEval-2014. The advantage of this dataset is that it is carefully cleaned from errors and also results of state-of-the-art systems are readily available for comparison. Table 3 demonstrates that in this dataset our system did not obtain top results. Still, LSTM performance is quite good (equivalent to 6th best result of 28 total participants).

Table 1. F-measure (proportional overlap) on SentiRuEval dataset, evaluated using 7-fold cross-validation

Mehod	SentiRuEval Restaurants dataset				SentiRuEval Cars dataset			
	Explicit	Implict	Fact	Macro average	Explicit	Implict	Fact	Macro average
Human Judge 1	69.1	58.7	33.0	53.6	—	—	—	—
Human Judge 2	65.0	62.3	27.0	51.4	—	—	—	—
CRF baseline	68.2	57.7	24.0	49.96	—	—	—	—
Logistic regression	54.0	43.0	3.0	33.3	70.1	75.4	15.2	53.6
MLP	64.5	53.6	18.2	45.3	75.8	82.2	34.8	64.2
Frame-NN	67.9	61.4	**26.1**	51.8	76.0	**83.0**	33.0	64.0
Simple RNN	68.4	58.5	20.0	48.9	75.2	81.3	30.1	62.2
Simple RNN augmented with one future word	68.9	60.0	25.3	51.4	75.8	82.0	31.4	63.1
Simple RNN augmented with one future word + dropout	71.1	56.0	20.1	49.06	76.0	82.1	24.3	60.8
Bidirectional RNN	69.8	61.2	19.1	50.3	76.1	81.5	32.1	63.2
Bidirectional LSTM	**73.5**	**64.3**	23.5	**53.76**	**77.0**	82.5	**36.3**	**65.3**

Table 2. F-measure on SentiRuEval Test dataset (according to SentiRuEval results)

Method	SentiRuEval Restaurants dataset				SentiRuEval Cars dataset			
	Proportional		Exact		Proportional		Exact	
	Explicit	All	Explicit	All	Explicit	All	Explicit	All
BRNN	67.2	52.2	57.5	64.5	71.7	70.4	61.7	59.9
LSTM	71.9	**60.0**	62.6	**66.8**	—	—	—	—
LSTM, Depth 2	—	—	—	—	**74.8**	**71.4**	65.1	63.0
Other systems best result	**72.8**	59.6	**63.1**	59.5	73.0	65.9	**67.6**	**63.6**

Table 3. Results on English SemEval ABSA Restaurant dataset (computed by us, using SemEval official metrics), reference results are taken from [Pontiki et al, 2014]

Method	F1 value
baseline	47.15
CRF with words and POS tags features	75.20
6th-best result	79.60

Method	F1 value
Top result	84.01
BRNN	76.20
LSTM	79.80

4.2. Sentiment polarity prediction task

Tables 4–6 summarize sentiment polarity results. Here more complex systems generally obtain superior results to simpler methodologies.

Using SentiRuEval-2015 official metrics we obtain second-best result in explicit aspect term polarity prediction on cars-dataset and third-result in restaurants dataset (unfortunately, results from our top systems were not included in official results due to errors that we made in data format. This error only became apparent after release of test sets and thus impossible to correct). Also, relatively poor results are partially explained by the fact that our system was optimized to all-term polarity prediction task, leading to suboptimal performance on explicit-term only task (information about official metrics were released by organizers with delay and we were not able to adapt all systems due to time and resource constraints). On English ABSA Restaurant dataset we obtain accuracy of 69.7, significantly below best results, but still reasonable.

Even through our results here are below top systems, they are reasonable good and have some theoretical value in demonstrating that exactly same architecture can be used both for sequence tagging and polarity prediction tasks. It also worth noting, that we used neither sentiment lexicon, nor special preprocessing steps for negation (we found that RNNs under certain conditions are capable to learn negation just from training data). Another important finding here that using hidden layer activations of RNNLM model as features instead of word vectors considerably improves overall system performance. Our hypothesis is that next-word prediction task of RNNLM includes the need to understand word dependencies—a knowledge that shown to be crucial in aspect-term polarity prediction task. This knowledge from unsupervised model can thus be leveraged by supervised RNN to enhance performance.

Table 4. Results on all-terms polarity prediction task on SentiRuEval dataset (F1 macro average on positive and negative classes and overall accuracy over all terms)

	Restaurants		Cars	
Method	Macro F1	Accuracy	Macro F1	Accuracy
TDNN N=3	61.0	57.4	55.2	56.2
RNN	63.1	59.2	57.1	57.1
BRNN	67.4	60.3	60.3	56.9
LSTM	70.2	61.1	62.4	58.0
LSTM + RNNLM features *	74.1	62.5	65.0	59.1

* Obtaining by using hidden layer activations of RNNLM

Table 5. Results on explicit-only terms polarity classification (according to SentiRuEval-2015 official results)

Method	Restaurants	Cars
BRNN	61.9	64.7
LSTM + RNNLM features	—	65.3
Top result	82.4	74.2

Table 6. Results for English terms polarity classification on ABSA Restaurants SemEval-2014 dataset (according to our evaluation metrics)

Method	Accuracy
Baseline	64.00
Sentiment lexica over dependency graphs *	69.50
BRNN	65.10
LSTM	69.70
Top result	82.92

* Value taken from [Wettendorf et al, 2015]

5. Conclusions

In aspect term extraction task recurrent neural networks models demonstrate excellent perfomance. On Russian SentiRuEval-2015 dataset our system obtained best result in extraction of all aspects terms according to proportional measure and best result in extraction of all aspect terms on cars dataset according to exact measure, while holding second-best result on restaurants dataset. On English SentEval-2014 dataset, we obtained reasonable good results, equivalent to 6th best known result on this dataset. From all RNN models, best results were obtained with deep bidirectional LSTM with 2 hidden layers.

For aspect term polarity predictions, we obtained second best result on SentiRuEval-2015 car dataset and third best result on SentiRuEval-2015 car restaurants dataset. We also obtained good results on all terms polarity prediction. To our knowledge, this is first time when LSTM models were applied to aspect term polarity prediction with reasonable good results.

Overall, our work demonstrates that RNN models are useful in aspect-based sentiment analysis and can be utilized for rapid prototyping and deployment of opinion mining systems in different languages.

Acknowledgments

Author want to thank Ekatertina Izotova for help with data format conversion, anonymous reviewers for helpful comments and SentiRuEval organizers for preparing and running evaluation and thus making this work possible.

References

1. *Blunsom, P., Grefenstette, E., & Kalchbrenner, N.* (2014). A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.
2. *Breck E., Choi Y., Cardie C.* (2007). Identifying expressions of opinion in context. In IJCAI, pp. 2683–2688.
3. *Brody S., Elhadad N.* (2010). An unsupervised aspect-sentiment model for online reviews. In Proceedings of NAACL, pp. 804–812, Los Angeles, California
4. *Choi Y., Cardie C.* (2010). Hierarchical sequential learning for extracting opinions and their attributes. In Proceedings of the ACL 2010 Conference Short Papers, pp. 269–274.
5. *dos Santos, C. N., & Gatti, M.* (2014). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland.
6. *Elman J.* (1990). Finding structure in time. Cognitive science, 14(2):179–211.
7. *Ganu, G., Elhadad, N., & Marian, A.* (2009, June). Beyond the Stars: Improving Rating Predictions using Review Text Content. In WebDB (Vol. 9, pp. 1–6).
8. *Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R.* (2012). Improving neural networks by preventing coadaptation of feature detectors. arXiv preprint arXiv:1207.0580
9. *Hochreiter, S., & Schmidhuber, J.* (1997). Long short-term memory. Neural computation, 9(8), 1735–1780.
10. *Irsoy O., Cardie C.* Opinion Mining with Deep Recurrent Neural Networks (2014). EMNLP, Doha, Qatar. pp. 720–728
11. *Johansson R., Moschitti A.* (2010). Syntactic and semantic structure for opinion expression detection. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pp. 67–76. Association for Computational Linguistics.
12. *Mesnil, G., He, X., Deng, L. & Bengio, Y.* (2013). Investigation of recurrent neural network architectures and learning methods for spoken language understanding. In INTERSPEECH pp. 3771–3775 : ISCA.
13. *Mikolov T., Karafi´at M., Burget L., Cernock´y J., Khudanpur S.* (2010). Recurrent neural network based language model. In INTERSPEECH, pp. 1045–1048.
14. *Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y.* (2013). How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026.
15. *Pontiki M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., & Manandhar, S.* (2014). Semeval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 27–35).
16. *Schuster M., Kuldip K. P.* (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681.
17. *Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D.* (2011, July). Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 151–161). Association for Computational Linguistics.

18. *Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C.* (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the conference on empirical methods in natural language processing (EMNLP) (Vol. 1631, p. 1642).
19. *Wenge R., Baolin P., Yuanxin O., Chao Li, Zhang X.* (2004) Structural information aware deep semi-supervised recurrent neural network for sentiment analysis. Frontiers of Computer Science, pp. 1–14, http://dx.doi.org/10.1007/s11704-014-4085-7
20. *Werbos, P. J.* (1990). Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10), 1550–1560.
21. *Wettendorf C., Jegan R., Korner A., Zerche J.* (2014) SNAP: A Multi-Stage XML-Pipeline for Aspect Based Sentiment Analysis In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 578–584

АНАЛИЗ ТОНАЛЬНОСТИ ТВИТОВ О ТЕЛЕКОММУНИКАЦИЯХ И БАНКАХ НА ОСНОВЕ МЕТОДА МАШИННОГО ОБУЧЕНИЯ В РАМКАХ SENTIRUEVAL

Тутубалина Е. В. (tutubalinaev@gmail.com)[1],
Загулова М. А. (mazagulova@stud.kpfu.ru)[1],
Иванов В. В. (nomemm@gmail.com)[1, 2],
Малых В. А. (valentin.malykh@phystech.edu)[3]

[1]Казанский (Приволжский) Федеральный Университет (КФУ), Казань, Россия
[2]Институт информатики, Академия наук Татарстана, Казань, Россия
[3]ИСА РАН, Москва, Россия

Ключевые слова: анализ тональности текстов, SentiRuEval, твиттер, классификация твитов

A SUPERVISED APPROACH FOR SENTIRUEVAL TASK ON SENTIMENT ANALYSIS OF TWEETS ABOUT TELECOM AND FINANCIAL COMPANIES

Tutubalina E. V. (tutubalinaev@gmail.com)[1],
Zagulova M. A. (mazagulova@stud.kpfu.ru)[1],
Ivanov V. V. (nomemm@gmail.com)[1, 2],
Malykh V. A. (valentin.malykh@phystech.edu)[3]

[1]Kazan Federal University (KFU), Kazan, Russia
[2]Institute of Informatics, Tatarstan Academy of Sciences, Kazan, Russia
[3]Institute for Systems Analysis RAS, Moscow, Russia

This paper describes a supervised approach for solving a task on sentiment analysis of tweets about banks and telecom operators. The task was articulated as a separate track in the Sentiment Evaluation for Russian (SentiRuEval-2015) initiative. The approach we proposed and evaluated is based on a Support

Vector Machine model that classifies sentiment polarities of tweets. The set of features includes term frequency features, twitter-specific features and lexicon-based features. Given a domain, two types of sentiment lexicons were generated for feature extraction: (i) manually created lexicons, constructed from *Pros* and *Cons* reviews; (ii) automatically generated lexicons, based on pointwise mutual information between unigrams in a training set.

In the paper we provide results of our method and compare them to results of other teams participated in the track. We achieved 35.2% of macro-averaged F-measure for banks and 44.77% for tweets about telecom operators. The method described in the paper is ranked second and fourth among 7 and 9 teams, respectively. The best SVM setting after tuning parameters of the classifier and error analysis with common types of errors are also presented in this paper.

Key words: sentiment analysis, sentirueval, twitter, social media, tweet sentiment classification

1. Introduction

Sentiment analysis has received much attention in recent years due to its capability to identify people's opinions about products, named entities, facts (or events), and companies. This field of study has become important, especially due to the rapid growth of microblogging services such as Twitter, in which people talk about their personal experiences.

The goal of this task is to determine whether a given tweet is positive, negative or neutral according to its influence on the reputation of telecom or financial company. It is generally difficult to implement traditional sentiment analysis of user reviews since tweets collection could be noisy and each message is limited in length and could contain misspelling, slang and short forms of words. There have been a large number of research studies in the area of sentiment classification of short informal texts that are well described in (Martínez-Cámara, 2014). State-of-the-art papers have applied various feature sets from traditional text classification features (e.g., ngrams, part of speech tags, stems) to twitter-specific features (e.g., emoticons, hashtags, abbreviations) to handle the task in supervised manner (Kiritchenko et al., 2014). Since sentiment analysis in English has been explored in depth, there are not much research on sentiment classification of users' reviews in Russian. The recent works have focused on solving a task on sentiment analysis during ROMIP sentiment analysis tracks in 2011–2013 (Chetviorkin and Loukachevitch, 2013; Kotelnikov and Klekovkina, 2012; Blinov et al., 2013; Frolov et al., 2013).

In this study we report our submission to the SentiRuEval task. The approach is based on a Support Vector Machine model. The set of features includes term frequency features i.e. word ngrams, character ngrams; twitter-specific features and lexicon-based features. Since lexicon-based features are the most useful features for sentiment classification of tweets in English, we generated two types of sentiment lexicons. These two types are: manually created lexicons, constructed from *Pros* and *Cons* reviews in a particular domain; automatically generated lexicons, based on pointwise

mutual information between unigrams in training set. We achieve 44.77% of macro-average F-measure of for tweets about telecommunications companies and 35.2% for banks domain, that give improvements of 26.54% and 22.53% in macro F1-measure over official baseline results, respectively.

The rest of the paper is organized as follows. In Section 2 we introduce related work on sentiment classification of short informal texts. In Section 3 we describe proposed classifiers with a set of text classification features and twitter-specific features. Section 4 presents results of experiments. Section 5 provides error analysis. Finally, in Section 6 we discuss the results and future extensions of our work.

2. Related Work

Extracting information from short informal texts, such as tweets or sms messages, has received much attention in sentiment analysis (Go, 2009; Kiritchenko et al., 2014; Sidorov et al., 2013), event detection (Sakaki et al., 2010), problem extraction (Gupta, 2013), sarcasm detection (Davidov et al., 2010) and public sentiment tracking (O'Connor et al., 2010). Traditional approaches of sentiment classification were based on the presence of words or emoticons that indicated positive or negative polarity (Turney, 2002; Taboada, 2010; O'Connor et al., 2010). State-of-the-art papers have implemented hybrid approaches based on the use of machine learning techniques and lexical resources such as sentiment lexicons (Mohammad et al., 2013; Zhu et al., 2014; Kiritchenko et al., 2014; Evert, 2014). Recent studies showed that important machine learning features are bag-of-words unigrams and bigrams, and the use of tweet syntax features (e.g., hashtags, retweets and links) can improve the classification results (Barbosa and Feng, 2010). In (Kiritchenko et al., 2014) authors showed the importance of determining the sentiment of words in the presence of negation. They used separate lexicons for terms in affirmative and negated contexts.

Much work in sentiment analysis involves the use of existing sentiment lexicons and generation of lexical resources capturing the sentiment of words (Martínez-Cámara, 2014). The generation of lexicons range from manual approaches of annotating lexicons to fully automated approaches. In (Evert, 2014) authors used manual extension of existing sentiment lexicons and dictionaries of emoticons and internet slang. In (Mohammad et al., 2013) authors created automatically generated hashtag lexicon estimating sentiment scores for terms based on pointwise mutual information between terms and tweets with polarities. Inspired by these works, that describe supervised methods top-ranked in the SemEval-2014 task about sentiment analysis of tweets in English, we decided to create sentiment lexicons in similar way.

Sentiment analysis of texts in Russian is less studied. In (Chetviorkin and Loukachevitch, 2013) authors describe the first open sentiment task about sentiment classification of users reviews in Russian. Supervised methods, based on SVM classifier in a combination of manual or automatic dictionaries or rule-based systems, are top-ranked for reviews about movies, books, and digital cameras in the task. In (Frolov et al., 2013) authors proposed an approach based on special dictionaries and fact semantic filters in sentiment analysis of user reviews about books. In (Blinov et al.,

2013) authors used manual emotional dictionaries for each of three domains and showed benefits of machine learning method over lexical approach for user reviews in Russian. They reported that it was difficult to select particular machine learning method with the best results in all review domains.

3. Twitter-based Sentiment Classification

The task determines whether each tweet about a telecommunication companies (ttk) or banks contains a positive, negative, or neutral sentiment. We applied a machine-learning approach, based on bag-of-words model and a set of twitter-specific, lexicon-based features that are described in section 3.3.

The following examples illustrate situations in which different types of classification features appear in a tweet. Tweets such as "Лучи дикой ненависти вашей организации, ГОРИТЕ В АДУ *бешусь*" ("Sending rays of wild hatred to your organization, BURN IN HELL *rage*") contain strong negative polarities with regards to words with all characters in upper case. Tweets such as "Почему у дебетовой карты списали деньги просто так?!" ("Why was money from my debit card taken out with no reason?!") and "Сеть прыгает из E в 3G и обратно каждые 5 минут ((" ("Network shifts from E to 3G every 5 minutes ((") do not contain any positive and negative words. Therefore, a human annotator detects negative sentiment in each tweet with regards to the context of the tweet and whether the last symbols are emoticons, exclamation or question marks. Emoticons indicate positive or negative sentiment in short tweets, e.g. "@sberbank всё спасибо, готово :)" ("@sberbank thank you, it is done :)") and "сбербанк продлил рассмотрение дела до 160 дней :(" ("Sberbank has prolonged consideration of the case till 160 days :("). Complex sentiment analysis in tweets such as "Проехать полгорода и узнать, что карта в другом из банков. Всегда мечтал .__." ("Crossed half the city to hear that my card is in another bank. I have always dreamed .__.") shows that some emoticons present sarcasm, which means that the opposite polarity of the positive word *мечтал* (*dreamed*) is denoted in the tweet. Presence of twitter-specific features such as URL or a retweet indicate to neutral context of tweets about news or informal messages, e.g. "mts коннект драйвер для android http://t.co/J3I5SNZuKM" ("mts connect driver for android *URL*") and "RT @Anna_Anna29: в билайне как узнать свой номер http://t.co/FpDZtLbdMZ" ("RT @Anna_Anna2: how to know your number in Beeline *URL*").

In the following examples we consider the use of sentiment lexicons, created manually and automatically. Manually created sentiment lexicons have been successfully applied in sentiment analysis in traditional approaches that detect whether a message contains positive or negative sentiment (Turney, 2002). The tweets such as "хреновый интернет, отвратительная работа с клиентами. Никогда не связывайтесь с этой шайкой" ("the lousy Internet, disgusting operation with clients. Never communicate with this gang") and "МТС пожелали хорошего дня, даже не попытались ничего продать. Уверовал в добро" ("MTS wished good day to me, didn›t even try to sell anything. I have believed in good") contain mention of domain-independent sentiment words like *отвратительный (disgusting)* and *хороший (good)*. Many tweets require deeper sentiment analysis due to difficult context

of messages, e.g. the negative tweets "к вашему интернету хочется приложить подорожник" ("there is a wish to put a plantain to your internet") or "Билайн, отдай мне мой интернет" ("Beeline, give me my internet"). For these reasons, other sentiment lexicon is automatically created to cover such cases.

We tested three different learning algorithms: Naive Bayes, logistic regression (MaxEnt) and Support Vector Machine model (SVM). The squared euclidean norm L2 is selected as the standard regularizer for linear models. Based on the results obtained on the training sets we select SVM with default parameters[1] for tweet classification in banks domain.

3.1. Two Types of Sentiment Lexicons

We explore two main methods to construct sentiment lexicons: manual and automatic.

In the manual method we collected user rated reviews from otzovik.com: 3357 reviews about banks and 1928 reviews about telecom companies. To make corpus more accurate, we included only *Pros* reviews into positive corpus and *Cons* reviews into negative corpus. *Pros* (*Преимущества*) and *Cons* (*Недостатки*) are parts of a review that describe strong reasons why an author of the review likes or dislikes the product aspect, respectively. For each domain we selected the top K adverbs, adjectives, verbs, and nouns which have the highest frequencies in each corpus. Then we reduced noun words, expressing explicit aspects in a user review of particular domain due to neutral polarity of these aspects (e.g., *связь* (*connection*), *услуга* (*service*), *платеж* (*payment*), *скорость* (*speed*), *сотрудник* (*employee*)). In addition, we reduced the most common adjectives (e.g., *российский* (*russian*), *большой* (*big*), *абонентский* (*subscriber*)) and verbs expressing an action (e.g., *использовать* (*use*), *написать* (*write*), *подключать* (*connect*)). For each word we added other word forms. The dictionary consists of about 139 positive and 131 negative words in banks domain. The dictionary consists of about 68 positive and 168 negative words in telecom companies domain.

Following Mohammad et al. (2013) and other state-of-art approaches, automatically generated lexicons are based on sentiment score for each term w in the training test:

$$score\,(w) = PMI\,(w, pt) - PMI\,(w, nt)$$

$$PMI\,(w, pt) = log_2 \frac{p\,(w, pt)}{p\,(w) \times p\,(pt)}$$

where *PMI* is pointwise mutual information, *pt* denotes positive tweets, *nt* denotes negative tweets, *p (w)*, *p (pt)*, and *p (w, pt)* are probabilities of *w* occurs in positive corpus. The words with strong sentiment polarities have statistically significant difference between *PMI (w, pt)* and *PMI (w, nt)* in contrast to neutral words. For example, the pair of values (*PMI (w, pt)*, *PMI (w, nt)*) computed over the tweets in banks domain

[1] We have used the scikit-learn library in Python.

equals (−0.8016, 0.1450) for the neural word *еда* (*food*); (−15.2438, 1.5649) for the negative word *ущерб* (*loss*) and (2.1839, −19.2026) for the positive word *выгодный* (*profitable*). Since tweets contain low-frequency noisy words, we ignored terms that occurred less than three times in the training set.

3.2. Preprocessing for Short Informal Texts

Since raw tweets are usually informal and very noisy, the following preprocessing steps are performed. User mentions are normalized to @username. The morpho-syntactic analyzer[2] is applied to replace the words in the tweet with the base forms. We define negated context as a part of tweet between a negation (e.g., a particle *не* (*no*), a predicative expression *нет* (*not*)) word and a punctuation mark. Words with related negations (the words after negations) are modified in conjunction with the negation tag "neg_". We identify emoticons and replace them with corresponding sentiment expressions[3] (e.g., we replace ':-)' with *happy*, 'o_0' with *surprise* and ' ;-]' with *wink*).

3.3. Classification Features for Sentiment Classification of Tweets

Each tweet is represented as a feature vector; brief descriptions of the features that we use are presented below:

- **word n-grams**: unigrams (single words) and bigrams (multiword expressions) extracted from a tweet are used as the features. Features with document frequency greater than two are selected.
- **character n-grams**: lowercased characters n-grams for $n = 2, \ldots, 4$ with document frequency greater than two were considered for feature selection.
- **all-caps words**: the feature counts the number of words which contain all capitalized characters. Abbreviations of companies (e.g., *МТС* (*MTS*), *ВТБ* (*VTB*)) are excluded.
- **punctuation**: the features count the number of marks in sequences of exclamation marks, question marks, or a combination of these marks and the number of marks in contiguous sequences of dots. Sequences that consisted of more than one mark are considered for feature selection.
- **last symbol**: a binary feature indicates whether the last symbol of a tweet is an exclamation mark or a bracket.
- **emoticons**: four features are extracted: the number of positive emoticons; the number of negative emoticons; two binary features that indicate whether a last symbol of a tweet is a positive or negative emoticon, respectively.

---

[2] We have used Mystem tool, url: https://tech.yandex.ru/mystem/

[3] We have used some sentiment expressions from http://en.wikipedia.org/wiki/List_of_emoticons

- **twitter-specific features**: three binary features that indicate whether a tweet contains mentions of a twitter user, a retweet, and a presence of URL.
- **lexicon-based features:** for each of the two generated lexicons, the features are calculated as follows:
  - for the manual created lexicon we count the number of positive sentiment words, negative sentiment words. Sentiment words with negations change the sentiment polarity, e.g. a positive word with a negation suffix consider as a negative word.
  - for the automatically created lexicon four features are added: the count of words with non-zero scores; the sum of the words' sentiment scores normalized by words' count; the maximal sentiment score and minimum sentiment score in a tweet. Sentiment words with negations shift the sentiment score towards the opposite polarity.

4. Experimental Results

We used the training set of 5,000 annotated tweets for each domain provided for the SentiRuEval task. The final number of tweets in the testing collection is 4,549 tweets about banks and 3,845 tweets about telecom companies.

The official results obtained by our classifiers on the testing set are presented in Table 1. The table shows the official baseline results and the results of the method, ranked first according to macro-average F-measure as the main quality measure in the task (Loukachevitch et al., 2015). Macro-average F-measure is calculated as the average value between F-measure of the positive class and F-measure of the negative class. The classifier was trained to predict all three classes (positive, negative, and neutral), but this macro-averaged measure does not consider any correctly classifying neutral tweets. Our method is second among 7 teams with 14 runs in banks domain. The method is ranked fourth among 9 teams and fifth among 19 runs in telecom companies domain. The best approach has a 0.007% improvement in macro F1-measure over our approach in banks domain.

Table 1. Performance metrics in tweet classification task in two domains: telecom companies and banks

	telecom companies		**banks**	
	micro F	**macro F**	**micro F**	**macro F**
Best	0.536	0.488	0.343	0.359
Our approach	0.528	0.448	0.337	0.352
Official baseline	0.337	0.182	0.238	0.127

We also present feature ablation experiments on the testing set, removing one each individual feature category from the full set. Table 2 shows the results of the ablation experiments, each row shows macro-average precision, macro-average recall,

and macro-average F-measure, calculated as the average value between corresponding measures of the positive and the negative classes. The most effective features are word n-grams for tweets about telecom companies. The most effective features are based on character n-grams and emoticons in banks domain. The method also archives an improvement of 0.021% in F-measure after reducing word n-grams in banks domain and an improvement of 0.041% in F-measure after reducing word automatic lexicons in ttk domain. These improvements could be caused by a dynamic context of tweet messages about companies. The tweets of the training set were published in 2014, the tweets of the testing set were written in 2013.

Table 2. Experimental Results for the ablation experiments in two domains

	telecom companies (ttk)			**banks**		
	macro P	**macro R**	**macro F**	**macro P**	**macro R**	**macro F**
All features	0.443	0.471	0.447	0.538	0.279	0.352
w/o character n-grams	0.447	0.413	0.405	0.444	0.233	0.301
w/o emoticons	0.413	0.450	0.406	0.489	0.274	0.335
w/o both lexicons	0.419	0.553	0.475	0.496	0.276	0.337
w/o last symbol	0.458	0.379	0.390	0.509	0.274	0.340
w/o lexicon (manual ver.)	0.379	0.505	0.432	0.516	0.270	0.340
w/o lexicon (automatic v.)	0.427	0.569	0.488	0.426	0.292	0.343
w/o all-caps words	0.446	0.447	0.436	0.498	0.293	0.349
w/o punctuation	0.429	0.429	0.412	0.522	0.286	0.350
w/o twitter syntax features	0.447	0.441	0.443	0.491	0.289	0.351
w/o word n-grams	0.390	0.412	0.373	0.507	0.316	0.373

We also analyzed the significance of SVM tuning to our method. After shifting SVM's regularized regression method to elastic net that linearly combines the L1 and L2 penalties and the regularization term's alpha to 0.0001, the classifier had the improvements of 4–5% in macro F1-measures over our results with SVM's default parameters in both domains. The tuned classifier achieves a macro-average F-measure of 39.46% for banks domain and of 50.6% for tweets about telecommunications companies. The results show that careful tuning of the machine learning algorithm could obtain much better results.

5. Error Analysis

After error analysis we identify the following types of most frequent errors in tweet classification:

- misspelling and difficulty with transliteration of English text into Russian
- multiple hashtags

- emotional discussion of neutral topics
- insufficient size of sentiment lexicons (presence of out-of-lexicon words in the testing set)

From Table 3 shows that most of the errors are caused by insufficient information about context in positive or negative tweets about companies.

Table 3. Error types distribution

	Misspelling and transliteration	**Multiple hashtags**	**Emotional discussion**	**Insufficient size of sentiment lexicons**
telecom companies	20.40%	8%	14.90%	43%
banks	9%	1%	11%	64%

Tweets such as "Билайну труба короче" ("Beeline›s game›s over") contain hidden negative meaning like "game›s over" with the word "труба" ("a pipe"). Negative tweets such as "Самый безалаберный банк!" ("The most disorganized bank!") are missclassified due to low-frequency words like "безалаберный" that are not contained in the training set nor created lexicons.

We haven›t applied error correlation for cases of orthographic errors like *ацтой* (*rubbish*) and *чорд* (*damn*), while the correct spellings of these words are included in manually created lexicons. Tweets such as "Билайн. Дисконнектинг пипл." ("Beeline. Disconnecting people.") with transliterated words with strong negative polarity in English were misclassified as neutral. The analysis shows that misspelling caused less errors to tweets than elongated, transliterated words, and presence of asterisk (star symbol) in foul language words.

Hashtags such as *#отстойсвязь* (*#yourconnectionsucks*), *#мтсумри* (*#mtsdie*), *#люблюего* (*#loveit*) contain strong sentiment orientation. 8% of errors in telecommunications would be eliminated by splitting hashtags into words and then calculated the sentiment scores of hashtags.

Fourth type the errors is related to neutral tweets about telecom companies or banks, that contain positive or negative polarity about other topics (e.g., tweets about a *company's dress code,* friendly conversation or flirting with a company's worker*). Other type of such tweets is* a tweet describing some daily company's event: "Матч штабквартиры Вымпелком — Сибирь. Пока ведем!!! :)" ("Match of Vympelcom›s headquarters Vs Siberia. We›re winning!!! :)"). In all these cases the tweet about the company is neutral. Our classifiers haven›t considered such cases that affect up to 11% of errors about bank tweets, and 14.9% of errors in telecommunication tweets.

6. Conclusion

In this paper we described a supervised method for sentiment classification of financial or telecom twitter data with an emphasis on consumer experience. The proposed method exploits Support Vector Machines with term frequency features, twitter-specific

features and lexicon-based features. Given a tweet the lexicon-based features were generated by checking whether a word is in sentiment lexicons, that were created both automatically and manually from user reviews. In order to produce an automatically created lexicon, we used pointwise mutual information to calculate sentiment score and associate each word from a training set with a proper sentiment class.

We demonstrated that by using these features, classification performance increases from a baseline macro-averaged F-measures of 0.265 to 0.447 for telecoms and of 0.225 to 0.352 for banks. We plan to create large corpora of positive and negative tweets for the sake of improvement of the classifiers with automatically created lexicons.

Acknowledgments

This work was funded by the subsidy of the Russian Government to support the Program of competitive growth of Kazan Federal University and supported by Russian Foundation for Basic Research (RFBR Project 13-07-00773).

References

1. *Barbosa L., Feng J.* (2010), Robust sentiment detection on Twitter from biased and noisy data, Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, pp. 38–42
2. *Blinov P., Klekovkina M., Kotelnikov E., Pestov O.* (2013), Research of lexical approach and machine learning methods for sentiment analysis, Computational Linguistics and Intellectual Technologies, Vol. 2(12), pp. 48–58.
3. *Chetviorkin I., Loukachevitch N.* (2013), Evaluating Sentiment Analysis Systems in Russian, ACL 2013, p. 14.
4. *Davidov D., Tsur O., 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.
5. *Evert S., Proisl T., Greiner P., Kabashi B.* (2014), SentiKLUE: Updating a Polarity Classifier in 48 Hours, SemEval 2014, Dublin, p. 551.
6. *Frolov A. V., Polyakov P. Yu., Pleshko V. V.* (2013), Using semantic filters in application to book reviews sentiment analysis, available at: www.dialog-21.ru/ digests/dialog2013/materials/pdf/FrolovAV.pdf
7. *Go A., Bhayani R., Huang L.* (2009), Twitter sentiment classification using distant supervision, CS224N Project Report, Stanford, pp. 1–12.
8. *Gupta N. K.* (2013), Extracting phrases describing problems with products and services from twitter messages, Computación y Sistemas, Vol. 17(2), pp. 197–206.
9. *Kiritchenko S., Zhu X., Mohammad S. M.* (2014), Sentiment analysis of short informal texts, Journal of Artificial Intelligence Research, Vol. 50, pp. 723–762.
10. *Kotelnikov, E. V., Klekovkina, M. V.* (2013), Sentiment analysis of texts based on machine learning methods [avtomaticheskij analiz tonal'nosti tekstov na osnove

metodov machinnogo obuchenija]. In Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog, pp. 753–762.
11. *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-2015, Moscow, pp. 3–9.
12. *Martínez-Cámara E., Martín-Valdivia M. T., Urena-López L. A., Montejo-Ráez A. R.* (2014), Sentiment analysis in twitter, Natural Language Engineering, Vol. 20(01), pp. 1–28.
13. *Mohammad S. M., Kiritchenko S., Zhu X.* (2013), NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets, Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR'13), Atlanta, p. 2
14. *O'Connor B., Balasubramanyan R., Routledge B. R., Smith N. A.* (2010), From tweets to polls: Linking text sentiment to public opinion time series, ICWSM-11, Barcelona, pp. 122–129.
15. *Sakaki T., Okazaki M., Matsuo Y.* (2010), Earthquake shakes Twitter users: real-time event detection by social sensors. Proceedings of the 19th international conference on World wide web, ACM, pp. 851–860.
16. *Sidorov G., Miranda-Jiménez S., Viveros-Jiménez F., Gelbukh A., Castro-Sánchez N., Velásquez F., Gordon J.* (2013), Empirical study of machine learning based approach for opinion mining in tweets, Advances in Artificial Intelligence, Vol. 7629, pp. 1–14.
17. *Taboada M., Brooke J., Tofiloski M., Voll K., Stede M.* (2011), Lexicon-based methods for sentiment analysis, Computational linguistics, Vol.37(2), pp. 267–307.
18. *Turney P. D.* (2002), Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, Proceedings of the 40th annual meeting on association for computational linguistics, Philadelphia, pp. 417–424.
19. *Wilson T., Wiebe J., Hoffmann P.* (2005), Recognizing contextual polarity in phrase-level sentiment analysis, Proceedings of the conference on human language technology and empirical methods in natural language processing, pp. 347–354.

ASPECT EXTRACTION AND TWITTER SENTIMENT CLASSIFICATION BY FRAGMENT RULES

Vasilyev V. G. (vvg_2000@mail.ru),
Denisenko A. A. (denisenko_alec@mail.ru),
Solovyev D. A. (dmitry_soloviev@bk.ru)

OOO «LAN-PROJECT», Moscow, Russia

The paper deals with approaches to explicit aspect extraction from user reviews of restaurants and sentiment classification of Twitter messages of telecommunication companies based on fragment rules. This paper presents fragment rule model to sentiment classification and explicit aspect extraction. Rules may be constructed manually by experts and automatically by using machine learning procedures. We propose machine learning algorithm for sentiment classification which uses terms that are made by fragment rules and some rule based techniques to explicit aspect extraction including a method based on filtration rule generation. The article presents the results of experiments on a test set for twitter sentiment classification of telecommunication companies and explicit aspect extraction from user review of restaurant. The paper compares the proposed algorithms with baseline and the best algorithm to track. Training sets, evaluation metrics and experiments are used according to SentiRuEval. As our future work, we can point out such directions as: applying semi-supervised methods for rule generation to reduce the labor cost, using active learning methods, constructing a visualization system for rule generation, which can provide the interaction process with experts.

Key words: fragment rules, sentiment classification, aspect extraction, opinion mining

1. Introduction

Opinion mining and sentiment extraction is an actively developing sub discipline of data mining and computational linguistics. A promising approach to automatic sentiment extraction is based on extraction of specific product features — aspects and on the determination of those polarities. Usually the problem is solved in three stages. At first aspects and those polarities are extracted. Then aspects gears to categories if they are predefined. Otherwise a set of aspects is clustered and representative aspects are selected. The final stage includes category polarity classification based on polarities of individual aspects.

In this paper we present a rule-based approach which exploits fragment rule model to explicit aspect extraction from user reviews and to sentiment classification of twitter messages. The main advantage of the approach is its good interpretability.

On the one hand, there is an opportunity to use expert knowledge in the model by means of constructing rules manually. On the other side, you can build the model automatically or get the interpretable model within a procedure, which includes interaction of an expert and a system.

In paper [7] approaches to sentiment classification of movie reviews are described. These approaches based on counting the number of the proposed positive and negative words and using Naive Bayesian classifier, maximum entropy classification, support vector machine. Using support vector machine raises accuracy to 82%. Another two methods of classification gives accuracy 75–80%. In paper [1] twitter sentiment classification based on support vector machine is described. The words, phrases and part of speech are used as features. The results shown in this paper are the same as results shown in the previous paper and stressed that using part of speech does not increased accuracy.

In paper [2] two approach to sentiment classification movie review. The first approach based on the number of positive and negative terms, intensification terms, and reverses the semantic polarity of a particular term. The second approach uses a machine learning algorithm, support vector machines. Using the first approach gives accuracy about 65–70%. Using the second approach raises accuracy to 85%. Combination the two approaches not increase accuracy.

In paper [3] authors propose approach to sentiment classification with polarity shifting detection. Polarity-shifted and polarity-unshifted sentences are used as features for classification based on support vector machine. This approach allows a few to improve the quality compared to the baseline.

In addition to the vocabulary and the vector approach for sentiment classification a number of papers propose special probabilistic models, for example, tree-based sentiment classification and using relationship between words [6]. Also, a number of papers the authors clearly define the rules of assessment texts. Particularly, in paper [7] different rule for determining the scope inverse word such as “no” are formulated. Thus, in the work on sentiment classification are used as standard methods for text classification, and modified methods, which take into account polarity shifted terms, the syntactic structure of sentences, the relationship between words.

In current paper approach to twitter sentiment classification based on features extracted by using fragment rules. Thus obtained features with proper setting of rules form the space of smaller dimension and have good descriptive power, as was shown in [10].

Aspect-based opinion mining has been widely researched. There are some known approaches to this task [4]: (1) frequency-based approach, (2) rule-based approach, (3) supervised learning techniques, (4) topic modelling techniques.

Frequency-based approach uses the fact that 60–70% of the aspects are explicit nouns [4]. It is argued that people writes reviews in aspect language because they also read other reviews and take the terminology. Rule-based approach uses the assumption that there is some kind of relation between aspects and polarities expressed in a text. A relation can be formalized by using rules. There is also a hybrid approach expressed in using rules for filtration of extracted noun phrases.

The problem may be considered as sequence labelling problem according to some suggested supervised machine learning methods. In particular, Hidden Markov Model

and Conditional Random Fields can be used. Topic modelling techniques use the natural assumption that topics of reviews are corresponding aspects.

In this paper, a rule-based approach to aspect extraction is proposed. There are two main rule models: grammar-based and fragment-based. Grammar models include the application of context-free grammars for example Tomita parser [8]. The other model is based on using special fragments from text and represents a number of operations under these fragments. A rule in this case is a declarative description of extracted information. Our model is an example of the last approach.

Due to the fact, that recall of aspect extraction can be achieved by using various dictionaries like thesaurus and domain-specific dictionaries, an important issue is improving precision. In this case, the improvements expressed in using special filtration mechanisms for extracted aspects. Here particularly fragment rules can be used. The purpose of participation in the track was testing fragment rule-based approaches to aspect extraction and tweet classification. In addition, we attempted to use methods for automatic fragment rule generation.

The remainder of the article is as follows. In section 2 a formal description of the fragment rule language and a description of proposed approaches is given. In section 3 obtained results are analyzed; a comparison with Baseline results and the best track results is given. Section 4 presents conclusion and future work.

2. Methods

2.1. Fragment rules model

In this work for describing text features and classification rules we used a mathematical model based on defining operations on sets of text fragments [9].

Let we have the text $D = (d_1, ..., d_n)$, where the $d_i \in T$ — single element of the text, $T = \{t_1, ..., t_m\}$ — the set of all elements, n — the length of the text, m — number of different elements of the text.

Definition 1

The set $\mathbb{F} = \{ (p, q) \mid 1 \le p \le q \le n \}$ will be called the set of all parts of the text length n. Fragments of the text will be called the single elements of the set $f = (f_l, f_r) \in \mathbb{F}$, that specify left f_l and right f_r border fragment (number of the first and last elements in fragment).

Definition 2

Let $f = (f_l, f_r) \in \mathbb{F}$ and $g = (g_l, g_r) \in \mathbb{F}$, then $|f| = f_r - f_l + 1$ — length of the fragment;
$g \sqsupset f$, if $g_l \le f_l \le f_r \le g_r$ and $f \ne g$ — inclusion relation;
$g \ll f$, if $g_l < f_l$ or $g_l = f_l \,\&\, f_r < g_r$ — order relation.

Definition 3

The set of fragments F will be called reduced, if there is no such $f \in F$, that $g \sqsupset f$. $R(F)$ denote reduced set of fragments based on the set F, R — reduce operation.

Definition 4

The distance between the fragments $f = (f_l, f_r) \in \mathbb{F}$ and $g = (g_l, g_r) \in \mathbb{F}$ is determined as follows:

$$d(f,g) = \begin{cases} g_l - f_r, f < g, \\ f_l - g_r, g < f, \\ g_l - f_r, g = f. \end{cases}$$

Definition 5

The result of the a rule Q for the text D is the set $F_Q \subset \mathbb{F}$, containing all of the fragment relevant this rule. If $F_Q \neq \emptyset$, then call the text D relevant rule Q.

Definition 6

Basic rules is a rule $Q = t, t \in T$ whose result is $F_Q = \{f_1, ..., f_l\}$ — reduced set of fragments, the elements that stand out in a single operation. Complex rule is a rule Q, which is obtained by performing operations on other rules $Q_1, ..., Q_k$.

Let us now determine the possible operations to build complex rules of Q from the basic rules $Q_1, ..., Q_k$.

Definition 7

$Q = Q_1 \nabla Q_2$ — binary operation OR, $F_Q \equiv R\left(F_{Q_1} \nabla F_{Q_2}\right)$,
$F_{Q_1} \nabla F_{Q_2} = \left\{f \in \mathbb{F} | \exists f_1 \in F_{Q_1}, f \sqsupset f_1 or\ \exists f_2 \in F_{Q_2}, f \sqsupset f_2\right\}$.

For example, the rule *good best quality* extract fragments relevant the appearance of these words in the text.

Definition 8

$Q = Q_1 \Delta_{n_1} Q_2$ — binary operation AND with limit on distance between fragments, $F_Q \equiv R\left(F_{Q_1} \Delta_{n_1} F_{Q_2}\right)$, $F_{Q_1} \Delta_{n_1} F_{Q_2} = \left\{f \in \mathbb{F} | \exists f_1 \in F_{Q_1} and\ \exists f_2 \in F_{Q_2}, that\ f \sqsupset f_1,\ f \sqsupset f_2 and\ d(f_1, f_2) \leq n_1\right\}$.

For example, the rule *beeline &4w LTE* extract fragments, in which distance between "*beeline*" and "*LTE*" less than 4 words. This operation can be used without any limits on the distance between the words.

Definition 9

$Q = Q_1 \square_{n_1,n_2} Q_2$ — binary operation of sequence with limit on distance between fragments, $F_Q \equiv R\left(F_{Q_1} \square_{n_1,n_2} F_{Q_2}\right)$, $F_{Q_1} \square_{n_1,n_2} F_{Q_2} = \{f \in \mathbb{F} | \exists f_1 \in F_{Q_1} and\ \exists f_2 \in F_{Q_2},\ that\ f_1 < f_2,\ d(f_1, f_2) > 0,\ f \sqsupset f_1, f \sqsupset f_2\ and\ n_1 \leq d(f_1, f_2) \leq n_2\}$.

For example, the rule *@Company: 3w (sale discount)* extract fragments, which after the name of the company at a distance of 3 words are words of "sale" or "discount". This operation can be used without any limits on the distance between the words.

Definition 10
$Q = \bowtie (Q_1, ..., Q_k)$ — multiple operation sequences of neighbouring elements (select of neighbouring fragments), $F_Q \equiv R\left(\bowtie \left(F_{Q_1}, \dots, F_{Q_k}\right)\right), \bowtie \left(F_{Q_1}, \dots, F_{Q_k}\right) = \{f \in \mathbb{F} | \exists f_i \in F_{Q_i}, i \in \overline{1,k}\colon\ f_i \prec f_{i+1},\ d(f_i, f_{i+1}) = 1, i \in \overline{1,k-1} and\ f \sqsupset f_i, i \in \overline{1,k}\}$.

For example, the rule "*(boss head director chief) (mts beeline megafon)"* extract phrases corresponding to different telecom executives.

Definition 11
$Q = Q_1 \between Q_2$ — binary operation finding the intersection of fragments, $F_Q \equiv \{f \in \mathbb{F} | f \in F_{Q_1} \wedge f \in F_{Q_2}\}$.

For example, the rule *[Chapter $SentBegin] extract words "Chapter"*, that are written in the beginning of the sentence.

Definition 12
$Q = Q_1 \lhd_{n_1,n_2}$ — unary operator imposes limitations on length of the fragment, $F_Q \equiv \{f \in F_{Q_1} | n_1 \le |f| \le n_2\}$.

For example, the rule *(beeline & mts) #IN #INTERVAL(2w/3w)* extract fragments containing specific words in length from 2 to 3 words.

To be able to construct rules include negation and conditional statements (when the presence of the expression is checked, but it is not included in the final fragment) are special variants of binary rules $\nabla, \Delta, \square, \bowtie, \between, \lhd,\ \Delta_{n_1}, \square_{n_1 n_2}$, in which one of the operands is considered negative or conditional. For example, $\square^{\rightarrow}_{n_1,n_2}$ is operator finding the sequence in which the second operand is taken from the negation; $\square^{\leftarrow}_{n_1,n_2}$ is operator finding the sequence in which the first operand is taken from the negation; $\square^{\rightarrow}_{n_1,n_2}$ — is operator finding the sequence in which the first operand is conditional. The rule $\square^{\rightarrow}_{n_1,n_2}$ defined as $Q = Q_1 \square^{\rightarrow}_{n_1,n_2} Q_2\ F_Q \equiv \{f \in F_{Q_1} | \exists! f_2 \in F_{Q_2}\colon f \prec f_2, 0 < n_1 \le d(f, f_2) \le n_2\}$.

For example, the rule *no ^:3 (good best quality)* extract the word "*good*", "*best*" and "*quality*" before which there is no word "no" at distance of three words.

#define command sets the named expression. In the pre-treatment rules text expression is substituted into the rule text. These expressions are used to avoid repeating elements in complex rules. *#set* command s used to set the saved variables. Unlike #define command at the first reference to the variable is made save search results and on subsequent calls text processing is not performed. To use named expressions or saved variables in the rule is necessary to use operators @ and @@.

For example, *#define Good (good best quality)* sets the named expression *Good,* which should be handled *@Good.*

2.2. Sentiment classification

For sentiment classification we used a hybrid approach which is based on combining rule-based feature extraction and classifier training by machine learning methods. Classifier induction includes training set pre-processing, feature extraction by using predefined set of fragment rules, training classifier by using selected machine learning methods.

Texts in the training set are pre-processed by using the following procedures:

1. Graphematical analysis (tokenization, sentence boundary detection, phonetic coding, word descriptors extraction).
2. Linguistic analysis (lemmatization, part of speech tagging, word sense disambiguation, collocation extraction, syntactic features extraction).
3. Low level indexes construction (inverted index of source word forms, inverted index of lemma word forms, inverted index of word descriptors).

The general scheme of the learning algorithm has the following form.

1. Building vector representation of texts by using the set of fragment rules.
2. Dimension reduction and feature weights calculation.
3. Training and evaluation of the classifier on the training set.

At the first step the predefined set of 100 special fragment rules are used for features extraction.

Example of fragment rule:
@@COND^:5((@@NEG^:5\s(@@INTENS^:5\s($Adj $Verb $Noun $Adv))) &5\s? @@OBJECT),

where @@COND — condition words (“if”), @@NEG — negative words, @@INTENS — intensive words (“very”, “far”, ”purely”), @@OBJECT — object (“mts”, “megafon”, “beeline”).

At the second step we used common methods for dimension reduction and feature weights calculation.

At the third step two classifiers are trained, one classifier for the positive class and one for the negative class. For classifier training we used our robust realization of the following standard machine learning methods:

1. Bayesian classifier based on multivariate Gaussian distribution (gmm),
2. K-nearest neighbours classifier (knn),
3. Von Mises-Fisher classifier (vmfs),
4. Roccio classifier (roccio),
5. Support vector machines classifier (svm).

Trained positive and negative classifiers are used for building the final decision rule of the following form:

$$d'(u) = \begin{cases} 1, d_{pos}(u) > d_{neg}(u) \mid d_{pos}(u) = d_{neg}(u) = 1, w_{pos}(u) > w_{neg}(u) \\ -1, d_{pos}(u) < d_{neg}(u) \mid d_{pos}(u) = d_{neg}(u) = 1, w_{pos}(u) < w_{neg}(u) \\ 0, d_{pos}(u) = d_{neg}(u) = 0 \end{cases}$$

where $d'(u) \in \{-1, 0, 1\}$ is the final decision rule, $d_{pos}(u) \in \{0, 1\}$ and $d_{neg}(u) \in \{0, 1\}$ is the decision rules for positive and negative class, $w_{pos}(u) \in [0, 1]$ and $w_{neg}(u) \in [0, 1]$

and degree of compliance positive or negative class (for probabilistic classifiers it is the probability assignment to the corresponding class, for svm it is the distance to corresponding hyperplane etc.), u — the set of features in the text.

2.3. Rule-based explicit aspect extraction

There are two types of aspects defined in aspect-based opinion mining: explicit and implicit. Explicit aspects are concepts that explicitly mentioned in a sentence. Implicit aspects are expressed indirectly. This section proposes a number of approaches to explicit aspect extraction based on fragment rules. Preliminary let $A = \{a_1, ..., a_n\}$ be a set off unique aspects extracted by experts and represented in the training set. Training set has been provide by SentiRuEval organizers [5].

Multiple operation OR

Basically for the purpose of explicit aspect extraction this kind of fragment rule can be used:

$$Q = Q_\nabla(a_1, a_2, ..., a_n),\ a_i \in A.$$

Here Q_∇ — is a rule, where operation OR acts as a connector between unique aspects. In fact, an appropriate set of fragments is extracted for each aspect. The result of the operation is a reduced united set of fragments.

Multiple operation OR with maximizing reduction

In the concerned case, the following situation may arise. Instead of a whole aspect, structural parts can be extracted. For example, there are three extracted aspects HOT, DISH, HOT DISH. A standard reduction method will delete the biggest fragment HOT DISH, and we'll have two aspects instead of one. In this regard, it was decided to modify the reduction method and to exclude fragments which are included in other fragments. Also it should be noted that neighbouring fragments may be one aspect. Therefore overlapping fragments and neighbouring fragments should be combined. As a result, fragments of the maximum length are extracted.

Rule-based filtration

Also it seems appropriate to use rule-based filtration for aspect extraction. The extraction algorithm constructed as follows. At first using aspects selected by experts fragments from an aspect to the nearest adjective are extracted. Then, the most common rules based on the extracted fragments (templates) are formed. Here in the feature space is defined previously. The generated rules are applied to filter the set of extracted candidate-aspects by counting support and removal of candidates with support below a threshold. As already mentioned, recall may be achieved by using appropriate dictionaries. In this case, the filtration process is necessary to improve precision. Definition of the context of some aspects allows to separate situations where the term is not an aspect.

Let (a_i) be a rule, a result is a set of fragments from the aspect a_i to the nearest adjective. The aspect extraction algorithm for each aspect selected by experts generates a set of aspect contexts $Q(a_i)$ by applying rule $Q(a_i)$ to the training set L.

Then the rule generation algorithm builds templates of these contexts. In each review candidate-aspects are extracted and filtered by using these templates. Finally, we have a set of extracted explicit aspects.

Algorithm2. Explicit aspect extraction with filtration

Input. A_L — set of aspects selected by experts
I — hierarchy of features,
L — train set,
R — test set

Output. A_T — extracted explicit aspects.

Step 1. For all $a_i \in A_L$
$GenerateRules(I, Q_L(a_i))$

Step 2. For all $r \in R$
For all $a_i \in A$
$A_T \leftarrow A_T \cup FilterAspects(Q_r(a_i))$

There are a number of classical algorithms for searching frequent item sets which used for generating rules such as *Apriori*, *FP-growth*, *Eclat*. One important difference between these algorithms is a method of data representation. Basically there are two approaches—horizontal and vertical representation. In the vertical representation it's necessary to have lists of fragments that match elements of a rule. In the horizontal representation each fragment corresponds to a set of rule elements. Vertical representation is more practical in case of the fragment model. In this context, it is possible to apply one of the known algorithms — Eclat [11]. Especially because support of rules is determined by the intersection of sets of fragments.

Rules of the form $Q_1 \square_{1,1} Q_2 \square_{1,1} \ldots \square_{1,1} Q_n$ are used for filtration. Searching of rules is based on a feature hierarchy. As elements of the hierarchy you may have parts of speech descriptors, single words, etc. Sequentially from the descriptor $Any (any word) a rule is expanding and specifying. A selection criterion is a degree of specificity of rules and a minimal support threshold. The specificity of the rules increases depending on a number of elements and their place in the hierarchy. The more elements and the lower the place of elements in the hierarchy then specificity is higher. In this case, the rules are eliminated with support below a threshold. As a result, every aspect is associated with set of rules. In such a way, filtration is done when there are only those candidate-aspects which match at least one rule.

3. Evaluation

3.1. Twitter sentiment classification

Used for teaching training set consisting of 3,846 tweets of telecommunications companies. Each company which was mentioned on Twitter rated on a scale {−1, 0, 1}.

Test set consists of 5,322 tweets about telecommunications companies. The objective of the testing was to include every mention of the company to one of three classes: positive, negative or neutral. Indicators macro *F*-measure and micro *F*-measure used to assess the quality. Test results are shown in Table 1. The table shows the best method, Baseline and 5 runs:

9_1 Bayesian classifier based on a mixture of multivariate normal distributions (*gmm*),
9_2 classifier k-nearest neighbours (*knn*),
9_3 Bayesian classifier based on the distribution of von Mises-Fisher (*vmfs*),
9_4 centroid classifier Roccio (*roccio*),
9_5 classifier based on support vector machines (svm).

Baseline refers all tweets to the most frequent class, in this case a negative. Used for teaching training set consisting of 3,846 tweets of telecommunications companies. Each company which was mentioned on Twitter rated on a scale {−1, 0, 1}.

Indicators macro *F*-measure and micro *F*-measure used to assess the quality [5].

Table 1. Evaluation of the quality of sentiment classification tweets

Algorithm	Macro *F*-measure	Micro *F*-measure
9_1 (gmm)	0,3158	0,3331
9_2 (knn)	0,2328	0,2626
9_3 (vmfs)	0,3305	0,3371
9_4 (roccio)	0,3310	0,3501
9_5 (svm)	0,3527	0,3765
Baseline	0,1823	0,3370
2_B	0,4829	0,5362

Evaluating the quality of classification are at Baseline micro *F*-measure and substantially higher macro *F*-measure. This can be explained feature Baseline and calculation rule micro and macro *F*-measure. Macro *F*-measure — is the average amount of standard *F*-measure that calculated separately for the three classes. Baseline algorithm has zero *F*-measure for two classes (positive and neutral), but *F*-measure negative class has a value of about 55%. By averaging the three classes *F*-measure is found to be 18%.Our algorithm solves these problems. The algorithm based on support vector machines shown best quality. The algorithm based on k-nearest neighbours showed the worst result. As we can see our result are comparable with result of other participants.

3.2. Explicit aspect extraction

Performance evaluation was made against the training set (gold standard), provided by organizers. The set consists of 202 annotated reviews in Russian. We used standard measures: precision, recall and F-measure. In official results the method based on multiple operation OR with maximizing reduction has identifier — 11.1.

Table 2. Evaluation results for explicit aspect extraction

	Strong demands			Weak demands		
Method	**P**	**R**	**F1**	**P**	**R**	**F1**
OR	49%	71%	58%	59%	72%	65%
Multiple operation OR with maximizing reduction [11.1]	51%	73%	60%	61%	74%	66%
Rule-basedfiltration	60%	64%	62%	66%	69%	67%
Baseline	55%	69%	61%	65%	70%	67%
[2.1] The best result/strong	72%	57%	63%	81%	62%	69%
[4.1] The best result/weak	55%	69%	61%	69%	79%	73%

In general, participants in the official track had comparable results. It turns out that the approach based on transferring aspects from the train set to the test set with normalization shows the same results as approaches used sophisticated models for training.

The results show that the modification of multiple OR operation generally contributes to the performance. It can be argued that maximizing reduction showed an advantage compared to minimizing reduction when there are only those fragments that contain no other. This reduction is applied in solving text classification tasks and offers advantages in terms of speed of execution of classification rules. In the future, different types of reduction can take the form of individual operations instead of using in default.

Application of rules in filtration also has a positive effect on the result, but there are a number of issues that require further study. Along with increasing precision recall decreases. To solve this problem it is advisable to consider other criteria of rule selection to find suitable experimental values of boundary parameters for rule specificity and support of candidate-aspects to achieve a minimum reduction of recall.

4. Conclusions and Future work

The paper deals with approaches to explicit aspect extraction and sentiment classification. The algorithm based on support vector machines shown best quality. The algorithm based on k-nearest neighbours showed the worst result. The results are at the level of the average results presented in sentiment analysis track. The algorithm

based on SVM using as features normalized lemma and syntactic links shown the best results on the track. In the efforts to extract the aspects we can say that the simplest approach shows comparable with the rest of the results. The use of filtering rules to improve the accuracy while reducing completeness. In this regard, it is necessary to separately evaluate the effect of boundary parameters on the result.

As our future work, we can point out such directions as: applying semi-supervised methods for rule generation to reduce the labor cost, using active learning methods, constructing a visualization system for rule generation, which can provide the interaction process with experts. Also expanding of the fragment rule model can give new expressive possibilities.

References

1. *Go A., Huang L., Bhayani R.* (2009), Twitter sentiment classification using distant supervision, CS224N Project Report, Stanford.
2. *Kennedy A., Inkpen D.* (2006), Sentiment Classification of Movie Reviews Using Contextual Valence Shifters, Computational Intelligence, vol. 22(2), pp. 110–125.
3. *Li S., Lee S. Y. M., Chen Y., Huang C.-R., Zhou G.* (2010), Sentiment Classification and Polarity Shifting, Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), Beijing, China, pp. 635–643.
4. *Liu B.* (2012), Sentiment Analysis and Opinion Mining, Morgan and Claypool Publishers.
5. *Loukachevitch N., Blinov P., Kotel'nikov P., Rubtsova Ju., Ivanov V., Tutubalina E.* (2015), SentiRuEval: Testing Object-Oriented Sentiment Analysis Systems in Russian.
6. *Nakagawa T., Inui K., Kurohashi S.* (2010), Dependency tree-based sentiment classification using CRFs with hidden variables, The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, USA, pp. 786–794.
7. *Pang B., Lee L., Vaithyanathan S.* (2002), Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP, Philadephia, Pennsylvania, USA, pp. 79–86.
8. *Tomita parser: https://tech.yandex.ru/tomita/*
9. *Vasilyev V. G.* (2011), Fragment extraction and text classification by logical rules [Klassifikatsija i vydelenie fragmentov v tekstah na osnove logicheskih pravil] Digital libraries: Advanced Methods and Technologies, Digital Collections RCDL'2011, Voronezh, pp. 133–139.
10. *Vasilyev V. G., Davidov S. U.* (2013), Sentiment classification by combined approach. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialog 2013", available at: www.dialog-21.ru/digests/dialog2013/materials/pdf/VasilyevVG.pdf
11. *Zaki M. J., Parthasarathy S., Ogihara M., Li W.* (1997), New Algorithms for Fast Discovery of Association Rules, Proceedings of the Third International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, USA, pp. 283–286.

Раздел III.

Анализ семантической близости

RUSSE: СЕМИНАР ПО ОЦЕНКЕ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ДЛЯ РУССКОГО ЯЗЫКА

Панченко А. (panchenko@lt.informatik.tu-darmstadt.de)
Дармштадтский технический университет, Дармштадт, Германия
Лувенский католический университет, Лувен, Бельгия

Лукашевич Н. В. (louk_nat@mail.ru)
МГУ имени М. В. Ломоносова, Москва, Россия

Усталов Д. (dau@imm.uran.ru)
Институт математики и механики им. Н. Н. Красовского Уральского отделения Российской академии наук, Екатеринбург, Россия;
NLPub, Екатеринбург, Россия

Паперно Д. (denis.paperno@unitn.it)
Университет Тренто, Роверето, Италия

Мейер К. М. (meyer@ukp.informatik.tu-darmstadt.de)
Дармштадтский технический университет, Дармштадт, Германия

Константинова Н. (n.konstantinova@wlv.ac.uk)
Университет Вулверхэмптона, Вулверхэмптон, Великобритания

Мероприятие RUSSE, представленное на конференции «Диалог 2015», посвящено исследованию систем определения семантической близости слов на русском языке. Для оценки таких систем предложено четыре подхода, основанных на человеческих оценках и классах семантических отношений. В мероприятии приняло участие 19 команд, приславших 105 моделей. Лучшие результаты показывают методы на основе обучения с учителем, сочетающие данные из разных источников. Несмотря на это, методы без учителя, такие как дистрибутивные модели, обученные на большом корпусе, демонстрируют сравнимые результаты. В статье приведено описание мероприятия RUSSE и приведены результаты проведённого эксперимента на существительных русского языка.

Ключевые слова: компьютерная лингвистика, лексическая семантика, меры семантической близости, семантические отношения, извлечение семантических отношений, синонимы, гиперонимы, когипонимы

RUSSE: THE FIRST WORKSHOP ON RUSSIAN SEMANTIC SIMILARITY

Panchenko A. (panchenko@lt.informatik.tu-darmstadt.de)
TU Darmstadt, Darmstadt, Germany
Université catholique de Louvain, Louvain-la-Neuve, Belgium

Loukachevitch N. V. (louk_nat@mail.ru)
Moscow State University, Moscow, Russia

Ustalov D. (dau@imm.uran.ru)
N. N. Krasovskii Institute of Mathematics and Mechanics,
Ural Branch of the RAS, Russia;
NLPub, Yekaterinburg, Russia

Paperno D. (denis.paperno@unitn.it)
University of Trento, Rovereto, Italy

Meyer C. M. (meyer@ukp.informatik.tu-darmstadt.de)
TU Darmstadt, Darmstadt, Germany

Konstantinova N. (n.konstantinova@wlv.ac.uk)
University of Wolverhampton, Wolverhampton, UK

The paper gives an overview of the Russian Semantic Similarity Evaluation (RUSSE) shared task held in conjunction with the Dialogue 2015 conference. There exist a lot of comparative studies on semantic similarity, yet no analysis of such measures was ever performed for the Russian language. Exploring this problem for the Russian language is even more interesting, because this language has features, such as rich morphology and free word order, which make it significantly different from English, German, and other well-studied languages. We attempt to bridge this gap by proposing a shared task on the semantic similarity of Russian nouns. Our key contribution is an evaluation methodology based on four novel benchmark datasets for the Russian language. Our analysis of the 105 submissions from 19 teams reveals that successful approaches for English, such as distributional and skip-gram models, are directly applicable to Russian as well. On the one hand, the best results in the contest were obtained by sophisticated supervised models that combine evidence from different sources. On the other hand, completely unsupervised approaches, such as a skip-gram model estimated on a large-scale corpus, were able score among the top 5 systems.

Keywords: computational linguistics, lexical semantics, semantic similarity measures, semantic relations, semantic relation extraction, semantic relatedness, synonyms, hypernyms, co-hyponyms

1. Introduction

A *similarity measure* is a numerical measure of the degree two given objects are alike. A *semantic similarity measure* is a specific kind of similarity measure designed to quantify the similarity of two lexical items such as nouns or multiword expressions. It yields high values for pairs of words in a semantic relation (synonyms, hyponyms, free associations, etc.) and low values for all other, unrelated pairs.

Semantic similarity measures proved useful in text processing applications, including text similarity, query expansion, question answering and word sense disambiguation [28]. A wide variety of measures were proposed and tested during the last 20 years, ranging from lexical-resource-based [31] to vector-based approaches, which in their turn evolved from Hyperspace Analogue to Language (HAL) by Lund and Burgess [24] to Latent Semantic Analysis (LSA) by Landauer and Dumais [20], topic models [12], Distributional Memory [2] and finally to neural network language models [26]. Many authors tried to perform exhaustive comparisons of existing approaches and developed a whole range of benchmarks and evaluation datasets. See Lee [22], Agirre et al. [1], Ferret [8], Panchenko [28], Baroni [4], Sahlgren [33], Curran [7], Zesch and Gurevych [38] and Van de Cruys [36] for an overview of the state-of-the-art techniques for English. A recent study of semantic similarity for morphologically rich languages, such as German and Greek, by Zervanou et al. [40] is relevant to our research. However, Russian is not considered in the latter experiment.

Unfortunately, most of the approaches to semantic similarity were implemented and evaluated only on a handful of European languages, mostly English. Some researchers, such as Krizhanovski [18], Turdakov [35], Krukov et al. [19] and Sokirko [34], worked towards adapting several methods developed for English to the Russian language. These efforts were, however, mostly done in the context of a few specific applications without a systematic evaluation and model comparison. To the best of our knowledge, no systematic investigation of semantic similarity measures for Russian was ever performed.

The very goal of the *Russian Semantic Similarity Evaluation* (*RUSSE*) shared task[1] is to fill this gap, conducting a systematic comparison and evaluation of semantic similarity measures for the Russian language. The event is organized as a competition where systems are calculating similarity between words of a joint, previously unseen gold standard dataset.

To this end, we release four novel test datasets for Russian and an open-source tool for evaluating semantic similarity measures[2]. Using this standardized evaluation methodology, we expect that each new semantic similarity measure for the Russian language can be seamlessly compared to the existing ones. To the best of our knowledge, *RUSSE* is the largest and most comprehensive evaluation of Russian similarity measures to date.

This paper is organized as follows: First, we describe previous shared tasks covering other languages. In Section 3, we outline the proposed evaluation methodology. Finally, Section 4 presents the key results of the shared task along with a brief discussion.

---

[1] http://russe.nlpub.ru

[2] https://github.com/nlpub/russe-evaluation/tree/master/russe/evaluation

2. Related Work

Evaluation of semantic similarity approaches can be fulfilled in various settings [3, 6, 21]. We identified three major research directions which are most related to our shared task.

The first strand of research is testing of automatic approaches relative to human judgments of word pair similarity. Most known gold standards for this task include the *RG* dataset [32], the *MC* dataset [27] and *WordSim353* [9]. These datasets were created for English. To enable similar experiments in other languages, there have been several attempts to translate these datasets into other languages. Gurevych translated the *RG* and *MC* datasets into German [13]; Hassan and Mihalcea translated them into Spanish, Arabic and Romanian [14]; Postma and Vossen [29] translate the datasets into Dutch; Jin and Wu [15] present a shared task for Chinese semantic similarity, where the authors translated the *WordSim353* dataset. Yang and Powers [37] proposed a dataset specifically for measuring verb similarity, which was later translated into German by Meyer and Gurevych [25].

Hassan and Mihalcea [14] and Postma and Vossen [29] divide their translation procedure into the following steps: disambiguation of the English word forms; selection of a translation for each word; additionally, translations were checked to be in the same relative frequency class as the source English word.

The second strand of research consists in testing of automated systems with respect to relations described in a lexical-semantic resource such as WordNet. Baroni and Lenci [3] stress that semantically related words differ in the type of relations between them, so they generate the BLESS dataset containing tuples of the form (w1, w2, relation). Types of relations include COORD (co-hyponyms), HYPER (hypernyms), MERO (meronyms), ATTRI (attributes—relation between a noun and an adjective expressing an attribute), EVENT (relation between a noun and a verb referring to actions or events). BLESS also contains, for each concept, a number of random words that were checked to be semantically unrelated to the target word. BLESS includes 200 English concrete single-word nouns having reasonably high frequency that are not very polysemous. The relata of the non-random relations are English nouns, verbs and adjectives selected and validated using several sources including WordNet, Wikipedia and the Web-derived ukWaC corpus.

The third strand of research evaluates possibilities of current automated systems to simulate the results of human word association experiments. The task originally captured the attention of psychologists, such as Griffiths and Steyvers [10–11]. One such task was organized in the framework of the CogALex workshop [30]. The participants received lists of five given words (primes) such as *circus*, *funny*, *nose*, *fool*, and *Coco* and were supposed to compute the word most closely associated to all of them. In this specific case, the word *clown* would be the expected response. 2,000 sets of five input words, together with the expected target words (associative responses) were provided as a training set to the participants. The test dataset contained another 2,000 sets of five input words. The training and the test datasets were both derived from the *Edinburgh Associative Thesaurus* (EAT) [16]. For each stimulus word, only the top five associations, i.e. the associations produced by the largest number of respondents, were retained, and all other associations were discarded.

3. Evaluation Methodology

In this section, we describe our approach to the evaluation of Russian semantic similarity measures used in the *RUSSE* shared task. Each participant had to calculate similarities between 14,836 word pairs[3]. Each submission was assessed on the following four benchmarks, each being a subset of these 14,836 word pairs:

1. **HJ.** Correlations with human judgments in terms of Spearman's rank correlation. This test set was composed of 333 word pairs.
2. **RT**. Quality of semantic relation classification in terms of average precision. This test set was composed of 9,548 word pairs (4,774 unrelated pairs and 4,774 synonyms and hypernyms from the *RuThes-lite* thesaurus[4]).
3. **AE**. Quality of semantic relation classification in terms of average precision. This test set was composed of 1,952 word pairs (976 unrelated pairs and 976 cognitive associations from the *Russian Associative Thesaurus*[5]).
4. **AE2.** Quality of semantic relations classification in terms of average precision. This test set was composed of 3,002 word pairs (1,501 unrelated pairs and 1,501 cognitive associations from a large-scale web-based associative experiment[6]).

In order to help participants to build their systems, we provided training data for each of the benchmarks (see Table 1). In case of the *HJ* dataset, it was only a small validation set of 66 pairs as annotation of word pairs is expensive. On the other hand, for the *RT*, *AE* and *AE2,* we had prepared substantial training collections of 104,518, 20,968, and 104,518 word pairs, respectively.

We did not limit the number of submissions per participant. Therefore, it was possible to present several models each optimised for a given type of semantic relation: synonyms, hypernyms or free associations. We describe each benchmark dataset below and summarize their key characteristics in Table 1.

Table 1. Evaluation datasets used in the RUSSE shared task

Name	Description	Source	#word pairs, test	#word pairs, train
HJ	human judgements	Crowdsourcing	333	66
RT	synonyms, hypernyms, hyponyms	RuThes Lite	9,548	104,518
AE	cognitive associations	Russian Associative Thesaurus	1,952	20,968
AE2	cognitive associations	Sociation.org	3,002	83,770

[3] https://github.com/nlpub/russe-evaluation/blob/master/russe/evaluation/test.csv

[4] http://www.labinform.ru/pub/ruthes/index.htm

[5] http://it-claim.ru/asis

[6] http://sociation.org

3.1. Evaluation based on Correlations with Human Judgments (HJ)

The first dataset is based on human judgments about semantic similarity. This is arguably the most common way to assess a semantic similarity measure. The *HJ* dataset contains word pairs translated from the widely used benchmarks for English: *MC* [27], *RG* [32] and *WordSim353* [9]. We translated all English words as Russian nouns, trying to keep constant the Russian translation of each individual English word. It is not possible to keep exact translations for all pairs that have an exact match between lexical semantic relations between the two languages because of the different structure of polysemy in English and Russian. For example, the pair *train* vs. *car* was translated as *поезд—машина* rather than *поезд—вагон* to keep the Russian equivalent of car consistent with other pairs in the datset. Evaluation metric in this benchmark is Spearman's rank correlation coefficient (ρ) between a vector of human judgments and the similarity scores. Table 2 shows an example of some relations from the *HJ* collection.

Table 2. Example of human judgements about semantic similarity (HJ)

word1	word2	sim
петух (cock)	петушок (cockerel)	0.952
побережье (coast)	берег (shore)	0.905
тип (type)	вид (kind)	0.852
миля (mile)	километр (kilometre)	0.792
чашка (cup)	посуда (tableware)	0.762
птица (bird)	петух (cock)	0.714
война (war)	войска (troops)	0.667
улица (street)	квартал (block)	0.667
…	…	…
доброволец (volunteer)	девиз (motto)	0.091
аккорд (chord)	улыбка (smile)	0.088
энергия (energy)	кризис (crisis)	0.083
бедствие (disaster)	площадь (area)	0.048
производство (production)	экипаж (crew)	0.048
мальчик (boy)	мудрец (sage)	0.042
прибыль (profit)	предупреждение (warning)	0.042
напиток (drink)	машина (car)	0.000
сахар (sugar)	подход (approach)	0.000
лес (forest)	погост (graveyard)	0.000
практика (practice)	учреждение (institution)	0.000

In order to collect human judgements, we utilized a simple crowdsourcing scheme that is similar to HITs in *Amazon Mechanical Turk*[7]. We decided to use a lightweight crowdsourcing software developed in-house due to the lack of native Russian

[7] https://www.mturk.com

speakers on popular platforms including *Amazon Mechanical Turk* and *CrowdFlower*[8]. The crowdsourcing process ran for 27 days from October 23 till November 19, 2014.

Firstly, we set up a special section on the *RUSSE* website and asked volunteers on Facebook and Twitter to participate in the experiment. Each annotator received an assignment consisting of 15 word pairs randomly selected from the 398 preliminarily prepared pairs, and has been asked to assess the similarity of each pair. The possible values of similarity were 0—not similar at all, 1—weak similarity, 2—moderate similarity, and 3—high similarity. Before the annotators began their work, we provided them with simple instructions[9] explaining the procedure and goals of the study.

Secondly, we defined two assignment generation modes for the word pairs: 1) a pair is annotated with a probability inversely proportional to the number of current annotations (*COUNT*); 2) a pair is annotated with a probability proportional to the standard deviation of annotations (*SD*). Initially, the *COUNT* mode has been used, but during the annotation process, we changed to mode to *SD* several times.

By the end of the experiment, we obtained a total of 4,200 answers, i.e. 280 submissions of 15 judgements. Some users participated in the study twice or more, annotating a different set of pairs each time. We used Krippendorff's alpha [17] with an ordinal distance function to measure the inter-rater agreement: $\alpha = 0.49$, which is a moderate agreement. The average standard deviation of answers by pair is $\bar{\sigma} = 0.62$ on the scale 0–3. This result can be explained primarily by two facts: (1) the participants were probably confusing “weak” and “moderate” similarity, and (2) some pairs were ambiguous or too abstract. For instance, it proved difficult for participants to estimate the similarity between the words «деньги» (“money”) and «отмывание» (“laundering”), because on the one hand, these words are associated, being closely connected within the concept of money laundering, while on the other hand these words are ontologically dissimilar and are indeed unrelated outside the particular context of money laundering.

3.2. Semantic Relation Classification of Synonyms and Hypernyms (RT)

This benchmark quantifies how well a system is able to detect synonyms and hypernyms, such as:

- автомобиль, машина, syn (car, automobile, syn)
- кошка, животное, hypo (cat, animal, hypo)

The evaluation dataset follows the structure of the *BLESS* dataset [3]. Each target word has the same number of related and unrelated source words as exemplified in Table 3. First, we gathered 4,774 synonyms and hypernyms from the *RuThes Lite* thesaurus [23]. We used only single word nouns at this step. These relations were considered positive examples. To generate negative examples we used the following procedure:

---

[8] http://www.crowdflower.com

[9] http://russe.nlpub.ru/task/annotate.txt

Input: P—a set of semantically related words (positive examples), C—text corpus[10].
Output: PN—a balanced set of semantic relations similar to BLESS [3] with positive and negative examples for each target word.

1. Start with no negative examples: N = {}.
2. Calculate PMI-based noun similarity matrix **S** from the corpus C, where similarity between words w_i and w_j:

$$s_{ij} = log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

$$= log \frac{\#w_i \text{ and } w_j \text{ coocurences in doc}}{\#word \text{ coocurences in doc}} * \frac{\#word \text{ occurences}}{\#\, w_i \text{ occurences}} * \frac{\#word \text{ occurences}}{\#\, w_j \text{ occurences}}$$

3. Remove similarities greater than zero from **S**: $s_{ij} = max(0, s_{ij})$.
4. For each positive example $<w_i, w_j> \in$ P:
   - Candidates are relations from **S** with the source word: $\{<w_i, w_j> : w_i = source, s_{ij} > 0\}$.
   - Rank the candidates by target word frequency $freq(w_j)$:
   - Add two top relations $<w_i, w_k>$ and $<w_i, w_m>$ to negative examples *N*.
   - Remove all relations $<*, w_k>$ and $<*, w_m>$ from consideration: $s_{ij} = 0$, for all i and $j \in \{k, m\}$.
5. Filter false negative relations with the help of human annotators. Each relation was annotated by at least two annotators. If at least one annotator indicates an error, remove this negative example from N.
6. The dataset PN is a union of positive and negative examples: $\{P \cup N\}$. Balance this dataset, so the number of positive and negative relations is equal for each source word.
7. Return PN.

The *Semantic Relation Classification* evaluation framework used here quantifies how well a system can distinguish related word pairs from unrelated ones. First, submitted word pairs are sorted by similarity. Second, we calculate the *average precision* metric [39]:

$$AveP = \frac{\sum P@r_r}{R}$$

Here r is the rank of each relevant pair, R is the total number of relevant pairs, and $P@r$ is the precision of the top-r pairs. This metric is relevant as it takes ranking into account; it corresponds to the area under the *precision-recall curve* (see Fig. 1).

It is important to note that average precision of a random baseline for the semantic relation classification benchmarks *RT*, *AE* and *AE2* is 0.5 as these datasets are balanced (each word has 50% of related and 50% of unrelated candidates). Therefore, *RT*, *AE* and *AE2* scores should not be confused with semantic relation extraction evaluation, a task where the ratio of related and unrelated candidates and the average precision are close to 0.0.

[10] In our experiments we used Russian Wikipedia corpus to induce unrelated words.

Table 3. Structure of the semantic relation classification benchmarks (RT, AE, AE2)

word1	word2	related
книга (book)	тетрадочка (notebook)	1
книга (book)	альманах (almanac)	1
книга (book)	сборничек (proceedings)	1
книга (book)	перекресток (crossroads)	0
книга (book)	марокко (marocco)	0
книга (book)	килограмм (kilogram)	0

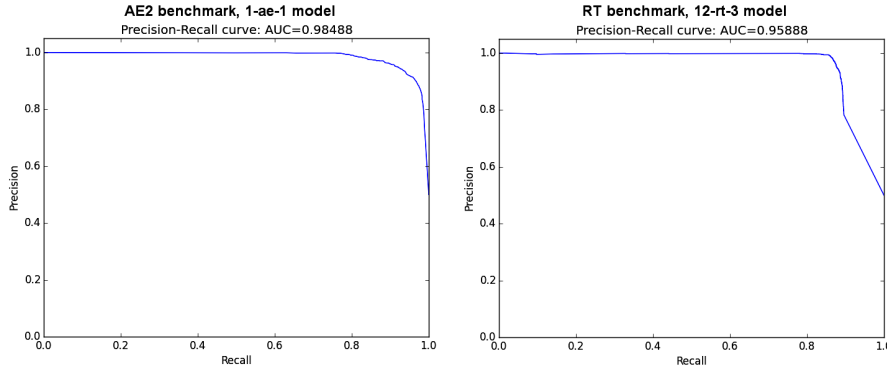


Fig. 1. Precision-recall curves of the best models on AE2 and RT datasets

3.3. Semantic Relation Classification of Associations (AE and AE2)

In the *AE* and *AE2* tasks, two words are considered similar if one is a cognitive (free) association of another. We used the results of two large-scale Russian *associative experiments* in order to build our training and test collections: the *Russian Associative Thesaurus*[11] (*AE*) and the Sociation.org (*AE2*). In an associative experiment, respondents were asked to provide a *reaction* to an input *stimulus*, e.g.:

- время, деньги, 14 (time, money, 14)
- россия, страна, 23 (russia, country, 23)
- рыба, жареная, 35 (fish, fried, 35)
- женщина, мужчина, 71 (woman, man, 77)
- песня, веселая, 33 (song, funny, 33)

The strength of an association is quantified by the number of respondents providing the same reaction. Associative thesauri typically contain a mix of synonyms, hyponyms, meronyms and other relations. Relations in such thesauri are often asymmetric.

[11] http://it-claim.ru/Projects/ASIS/

To build the test sets we gathered 976 and 1,501 associations respectively from the *Russian Associative Thesaurus* and the *Sociation.org*. At this step, we used the target words with the highest association value between stimulus and reaction. Similarly to the *RT* dataset, we used only single-word nouns. Negative word pairs i.e. semantically unrelated words, were generated with the procedure described in the previous section. In the same fashion as the *RT*, we use average precision to measure the performance on the *AE* and *AE2* benchmark datasets.

4. Results and Discussion

Initially, 52 groups registered for the shared task, which shows high interest in the topic. A total of 19 teams finally submitted at least one model. These participants uploaded 105 runs (1 to 17 runs per team). A table with the evaluation results of all these submissions is available online[12]. To make the paper more readable, we present only abridged results here. First, we removed near duplicate submissions. Second, we kept only the best models of each participant. If one model was better than another with respect to all four benchmarks then the latter was dropped.

Participants used a wide range of approaches in order to tackle the shared task including:

- distributional models with context window and syntactic context: participants 3, 10, 11, 17;
- network-based measures that exploit the structure of a lexical graph: participants 2, 19;
- knowledge-based measures, including linguistic ontologies, Wiktionary and Wikipedia relations: participants 8, 12;
- measures based on lexico-syntactic patterns: participant 4;
- systems based on unsupervised neural networks, such as *CBOW* [26]: participants 1, 5, 7, 9, 13, 15, 16;
- supervised models: participants 1, 2, 5, 15.

These methods were applied to corpora of different sizes and genres (see Table 4), including Wikipedia, the Russian National Corpus (RNC), RuWaC, a news corpus, a web crawled corpus, a Twitter corpus, and three collections of books (Google N-Grams, Lib.ru, and Lib.rus.ec). Detailed descriptions of some submissions are available in the proceedings of the Dialogue 2015 conference[13].

Table 6 in the appendix presents the top 10 models according to the correlations with human judgements (*HJ*). The best results were obtained by the model *5-rt-3*[14], combining corpus-, dictionary-, and morpheme-based features. As one may observe, systems building upon *CBOW* and *skip-gram* models [26] trained on a big corpus yielded good results in this task. On the other hand, the classical distributional context window model *17-rt-1* also managed to find its place among the top results. Finally, the recent *GloVe* model *16-ae-1* also proved successful for the Russian language.

---

[12] http://russe.nlpub.ru/results

[13] http://dialog-21.ru/dialog2015, see the Dialogue Evaluation on semantic similarity.

[14] here *5-rt-3* is a submission identifier, where the first number (5) denotes the number of participant

Table 4. Russian corpora used by participants

Corpus Name	Size, tokens
Russian Wikipedia	0.24 B
Russian National Corpus	0.20 B
lib.rus.ec	12.90 B
Russian Google N-grams	67.14 B
ruWaC	2.00 B
lib.ru	0.62 B

Table 5. 11 best models, sorted by the sum of scores. Each of the models is in top 5 of at least in one of the four benchmarks (HJ, RT, AE and AE2). Top 5 models are in bold font.

Model ID	HJ	RT-AVEP	AE-AVEP	AE2-AVEP	Method Description
5-ae-3	0.7071	0.9185	0.9550	0.9835	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, bigrams on the same corpus, synonym database, prefix dictionary, orthographic similarity
5-rt-3	**0.7625**	**0.9228**	0.8887	**0.9749**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, synonym database, prefix dictionary, orthographic similarity
1-ae-1	0.6378	**0.9201**	**0.9277**	**0.9849**	Desicion trees based on n-grams (Wikipedia titles and search queries), morphological features and Word2Vec
15-rt-2	0.6537	0.9034	**0.9123**	0.9646	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 500d vectors, window size 5, 3 iteration, min cnt 5)
16-ae-1	0.6395	0.8536	**0.9493**	0.9565	GloVe (100d vectors) on RuWac (lemmatized, normalized)
9-ae-9	**0.7187**	0.8839	0.8342	0.9517	Word2vec CBOW with window size 5 on Russian National Corpus, augmented with skip-gram model with context window size 20 on news corpus
17-rt-1	**0.7029**	0.8146	0.8945	0.9490	Distributional vector-based model, window size 5, trained on RUWAC and NRC, plmi-weighting
9-ae-6	**0.7044**	0.8625	0.8268	0.9649	Word2vec CBOW model with context window size 10 trained on web corpus
15-rt-1	0.6213	0.8472	**0.9120**	**0.9669**	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 100d vectors, window size 10, 1 iteration, min cnt 100)
1-rt-3	0.4939	**0.9209**	0.8500	**0.9723**	Logistic regression trained on synonyms, hyponyms and hypernyms on word2vec features with AUC maximization
12-rt-3	0.4710	**0.9589**	0.5651	0.7756	Applying knowledge extracted from Wikipedia and Wiktionary for computing semantic relatedness

Results of the *RT* benchmark (synonyms and hypernyms) are summarized in Table 7 in the appendix. The first place belongs to a knowledge-based model that builds upon Wiktionary and Wikipedia. Otherwise, all other models at the top are either based on standard *word2vec* tools or on a hybrid model that relies on *word2vec* embeddings.

Tables 8 and 9 list models that were able to successfully capture cognitive associations. The supervised models *5-ae-3* and *1-ae-1* that rely on heterogeneous features, including those from *CBOW/skip-gram* models, showed excellent results on both *AE* and *AE2* benchmarks. Like in the other tasks, the *word2vec*, *GloVe* and distributional context window models show very prominent results.

Interestingly, the systems are able to better model associations (top 10 submissions of *AE2* ranging from 0.96 to 0.99) than hypernyms and synonyms (top 10 submissions ranging from 0.85 to 0.96) as exemplified in Tables 8 and 10. Therefore, semantics that is mined by the skip-gram model and other systems is very similar to that of cognitive associations.

Again, we must stress here that the average precision of semantic relation *classification* presented in Tables 5–9 should not be confused with the average precision of the semantic relation *extraction,* which is normally much lower. Our evaluation schema was designed to learn relative ranking of different systems.

Finally, Table 5 lists the 11 most successful systems overall, ranked by the sum of scores. Each model in this table is among the top 5 of at least one of the four benchmark datasets. The best models either rely on big corpora (ruWaC, Russian National Corpus, lib.rus.ec, etc.) or on huge databases of lexical semantic knowledge, such as Wiktionary. While classical distributional models estimated on a big corpus yield good results, they are challenged by more recent models such as *skip-gram, CBOW* and *GloVe*. Finally, supervised models show that it is helpful in this context to adopt an unsupervised model for a certain type of semantic relations (e.g. synonymy vs. association) and to combine heterogeneous features for other types.

5. Conclusions

The *RUSSE* shared task became the first systematic attempt to evaluate semantic similarity measures for the Russian language. The 19 participating teams prepared 105 submissions based on distributional, network, knowledge and neural network-based similarity measures. The systems were trained on a wide variety of corpora ranging from the Russian National Corpus to Google N-grams. Our main contribution is an open-source evaluation framework that relies on our four novel evaluation datasets. This evaluation methodology lets us identify the most practical approaches to Russian semantic similarity. While the best results in the shared task were obtained with complex methods that combine lexical, morphological, semantic, and orthographic features, surprisingly, the unsupervised skip-gram model trained a completely raw text corpus was able to deliver results in top 5 best submissions according to 3 of the 4 benchmarks. Overall, the experiments show that common approaches to semantic similarity for English, such as *CBOW* or distributional models, can be successfully applied to Russian.

Semantic similarity measures can be *global* and *contextual* [5]. While this research investigated global approaches for Russian language, in future research

it would be interesting to investigate which contextual measures are most suited for languages with rich morphology and free word order, such as Russian.

Acknowledgements

This research was partially supported by the ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES), German Research Foundation (DFG) under the project JOIN-T, and Digital Society Laboratory LLC. We thank Denis Egorov for providing the *Sociation.org* data; Yuri N. Philippovich, Andrey Philippovich and Galina Cherkasova for preparing the evaluation dataset based on the Russian Associative Thesaurus. We would like to thank all those who participated in the crowdsourced annotation process during the construction of the dataset of human judgement. We thank Higher School of Economics' students who annotated unrelated word pairs used in the evaluation materials. Finally, we thank Prof. Iryna Gurevych and Ilia Chetviorkin for their help with the design of the experimental tasks and user interface.

References

1. *Agirre E., Alfonseca E., Hall K., Kravalova J., Paşca M., Soroa A.* (2009), A study on similarity and relatedness using distributional and wordnet-based approaches, Proceedings of NAACL-HLT 2009, Boulder, CO, USA, pp. 19–27.
2. *Baroni M., Lenci A.* (2009), One distributional memory, many semantic spaces. Proceedings of the EACL GEMS Workshop. Athens, Greece, pp. 1–8.
3. *Baroni M., Lenci A.* (2011), How we BLESSed distributional semantic evaluation, Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, Edinburgh, Scotland, pp. 1–10.
4. *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), Baltimore, MD, USA, pp. 238–247.
5. *Biemann C., Martin R.* (2013), Text: Now in 2D! a framework for lexical expansion with contextual similarity, Journal of Language Modelling, Vol. 1(1), pp. 55–95.
6. *Bullinaria J. A., Levy J. P.* (2012), Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD, Behavior Research Methods, Vol. 44(3), pp. 890–907.
7. *Curran J. R.* (2004), From distributional to semantic similarity, PhD thesis, University of Edinburgh, UK.
8. *Ferret O.* (2010), Testing semantic similarity measures for extracting synonyms from a corpus, Proceedings of LREC 2010, Valletta, Malta, pp. 3338–3343.
9. *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 10th International Conference on World Wide Web, Hong Kong, China, pp. 406–414.
10. *Griffiths T. L., Steyvers M.* (2002), A probabilistic approach to semantic representation, Proceedings of the 24th Annual Conference of the Cognitive Science Society, Fairfax, VA, USA, pp. 381–386.

11. *Griffiths T. L., Steyvers, M.* (2003), Prediction and semantic association, Advances in neural information processing systems 15, British Columbia, Canada, pp. 11–18.
12. *Griffiths T., Steyvers M., Tenenbaum J.* (2007), Topics in semantic representation, Psychological Review, Vol. 114, pp. 211–244.
13. *Gurevych I.* (2005), Using the Structure of a Conceptual Network in Computing Semantic Relatedness, Proceedings of the 2nd International Joint Conference on Natural Language Processing, Jeju Island, South Korea, pp. 767–778.
14. *Hassan S., Mihalcea R.* (2009), Cross-lingual Semantic Relatedness Using Encyclopedic Knowledge, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Vol. 3, Singapore, pp. 1192–1201.
15. *Jin P., Wu Y.* (2012), Semeval-2012 task 4: evaluating chinese word similarity, Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the Sixth International Workshop on Semantic Evaluation, Montréal, Canada, pp. 374–377.
16. *Kiss G., Armstrong C., Milroy R., Piper J.* (1973), An associative thesaurus of English and its computer analysis, The Computer and Literary Studies, Edinburgh University Press, Edinburgh, Scotland, UK, pp. 153–165.
17. *Krippendorff K.* (2013), Content Analysis: An Introduction to Its Methodology (Third Edition), SAGE, Thousand Oaks, CA, USA.
18. *Krizhanovski A. A.* (2007), Evaluation experiments on related terms search in Wikipedia, SPIIRAS Proceedings, Vol. 5, pp. 113–116.
19. *Krukov K. V., Pankova L. A., Pronina V. S., Sukhoverov V. S., Shiplina L. B.* (2010), Semantic similarity measures in ontology, Control Sciences, Vol. 5, pp. 2–14.
20. *Landauer T. K., Dumais S. T.* (1997), A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge, Psychological Review, Vol. 104(2), pp. 211–240.
21. *Lapesa G., Evert S.* (2014), A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions and Model Selection, Transactions of the Association for Computational Linguistics, Vol. 2, pp. 531–545.
22. *Lee L.* (1999), Measures of distributional similarity, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, MA, USA, pp. 25–32.
23. *Loukachevitch N. V., Dobrov B. V., Chetviorkin I. I.* (2014), RuThes-Lite, a Publicly Available Version of Thesauru of Russian Language RuThes, Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", Bekasovo, Russia, pp. 340–349.
24. *Lund K., Burgess C.* (1996), Producing high-dimensional semantic spaces from lexical co-occurrence, Behavior Research Methods, Vol. 28(2), pp. 203–208.
25. *Meyer C. M., Gurevych I.* (2012), To Exhibit is not to Loiter: A Multilingual, Sense-Disambiguated Wiktionary for Measuring Verb Similarity, Proceedings of COLING 2012: Technical Papers, Mumbai, India, pp. 1763–1780.
26. *Mikolov T., Chen K., Corrado G., Dean J.* (2013), Efficient Estimation of Word Representations in Vector Space, available at: http://arxiv.org/abs/1301.3781
27. *Miller G. A., Charles W. G.* (1991), Contextual correlates of semantic similarity, Language and Cognitive Processes, 6(1), pp. 1–28.

28. *Panchenko A.* (2013), Similarity measures for semantic relation extraction, PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium.
29. *Pennington J., Socher R., Manning, C. D.* (2014), Glove: Global vectors for word representation, Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP), Doha, Qatar, 1532–1543.
30. *Postma M., Vossen P.* (2014), What implementation and translation teach us: the case of semantic similarity measures in wordnets, Proceedings of Global WordNet Conference 2014, Tartu, Estonia, pp. 133–141.
31. *Rapp R., Zock M.* (2014), The CogALex-IV Shared Task on the Lexical Access Problem, Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon, Dublin, Ireland, pp. 1–14.
32. *Richardson R., Smeaton A., Murphy J.* (1994), Using WordNet as a Knowledge Base for Measuring Semantic Similarity between Words, Proceedings of AICS Conference, Dublin, Ireland.
33. *Rubenstein H., Goodenough J. B.* (1965), Contextual Correlates of Synonymy. Communications of the ACM, Vol. 8(10), pp. 627–633.
34. *Sahlgren M.* (2006), The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces, PhD thesis, Stockholm University, Stockholm, Sweden.
35. *Sokirko A.* (2013), Mining semantically similar language expressions for the Yandex information retrieval system (through to 2012) [Mayning blizkikh po smyslu vyrazheniy dlya poiskovoy sistemy Yandex (do 2012 goda)], available at: http://www.aot.ru/docs/MiningQueryExpan.pdf
36. *Turdakov D. Y.* (2010), Methods and software for term sense disambiguation based on document networks [Metody i programmnye sredstva razresheniya leksicheskoy mnogoznachnosti terminov na osnove setey dokumentov], PhD thesis, Lomonosov Moscow State University, Moscow, Russia.
37. *Van de Cruys, T.* (2010), Mining for Meaning: The Extraction of Lexicosemantic Knowledge from Text. PhD thesis, University of Groningen, Groningen, The Netherlands.
38. *Yang D., Powers D. M. W.* (2006), Verb Similarity on the Taxonomy of WordNet, Proceedings of GWC-06, Jeju Island, Korea, pp. 121–128.
39. *Zesch T., Gurevych I.* (2010), Wisdom of crowds versus wisdom of linguists–measuring the semantic relatedness of words, Natural Language Engineering, Vol. 16(1), pp. 25–59.
40. *Zhang E., Zhang Y.* (2009), Average Precision, Encyclopedia of Database Systems, Springer US, pp. 192–193.
41. *Zervanou K., Iosif E., Potamianos A.* (2014), Word Semantic Similarity for Morphologically Rich Languages, Proceedings of the Ninth International Conference on Language Resources and Evaluation, Reykjavik, Iceland, pp. 1642–1648.

Appendix 1. The Best Submissions of the RUSSE Shared Task

Table 6. 10 best models according to the HJ benchmark. Top 5 models are in bold font

Model ID	HJ	Method Description
5-rt-3	**0.7625**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, synonym database, prefix dictionary, orthographic similarity
9-ae-9	**0.7187**	Word2vec CBOW with window size 5 on Russian National Corpus, augmented with skip-gram model with context window size 20 on news corpus
5-ae-3	**0.7071**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, big-rams on the same corpus, synonym database, prefix dictionary, orthographic similarity
9-ae-6	**0.7044**	Word2vec CBOW model with context window size 10 trained on web corpus
17-rt-1	**0.7029**	Distributional vector-based model, window size 5, trained on RUWAC and NRC, plmi-weighting
15-rt-2	0.6537	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 500d vectors, window size 5, 3 iteration, min cnt 5)
16-ae-1	0.6395	GloVe (100d vectors) on RuWac (lemmatized, normalized)
1-ae-1	0.6378	Desicion trees based on n-grams (Wikipedia titles and search queries), morphological features and Word2Vec
15-rt-1	0.6213	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 100d vectors, window size 10, 1 iteration, min cnt 100)
1-rt-3	0.4939	Logistic regression trained on synonyms, hyponyms and hypernyms on word2vec features with AUC maximization
12-rt-3	0.4710	Applying knowledge extracted from Wikipedia and Wiktionary for computing semantic relatedness

Table 7. 10 best models according to the RT benchmark. Top 5 models are in bold font

Model ID	RT-AVEP	Method Description
12-rt-3	**0.9589**	Applying knowledge extracted from Wikipedia and Wiktionary for computing semantic relatedness
5-rt-3	**0.9228**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, synonym database, prefix dictionary, orthographic similarity
1-rt-3	**0.9209**	Logistic regression trained on synonyms, hyponyms and hypernyms on word2vec features with AUC maximization
1-ae-1	**0.9201**	Desicion trees based on n-grams (Wikipedia titles and search queries), morphological features and Word2Vec
5-ae-3	**0.9185**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, big-rams on the same corpus, synonym database, prefix dictionary, orthographic similarity
15-rt-2	0.9034	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 500d vectors, window size 5, 3 iteration, min cnt 5)
9-ae-9	0.8839	Word2vec CBOW with window size 5 on Russian National Corpus, augmented with skip-gram model with context window size 20 on news corpus
9-ae-6	0.8625	Word2vec CBOW model with context window size 10 trained on web corpus
16-ae-1	0.8536	GloVe (100d vectors) on RuWac (lemmatized, normalized)
15-rt-1	0.8472	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 100d vectors, window size 10, 1 iteration, min cnt 100)
17-rt-1	0.8146	Distributional vector-based model, window size 5, trained on RUWAC and NRC, plmi-weighting

Table 8. 10 best models according to the AE benchmark. Top 5 models are in bold font

Model ID	AE-AVEP	Method Description
5-ae-3	**0.9550**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, bigrams on the same corpus, synonym database, prefix dictionary, orthographic similarity
16-ae-1	**0.9493**	GloVe (100d vectors) on RuWac (lemmatized, normalized)
1-ae-1	**0.9277**	Desicion trees based on n-grams (Wikipedia titles and search queries), morphological features and Word2Vec
15-rt-2	**0.9123**	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 500d vectors, window size 5, 3 iteration, min cnt 5)
15-rt-1	**0.9120**	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 100d vectors, window size 10, 1 iteration, min cnt 100)
17-rt-1	0.8945	Distributional vector-based model, window size 5, trained on RUWAC and NRC, plmi-weighting
5-rt-3	0.8887	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, synonym database, prefix dictionary, orthographic similarity
1-rt-3	0.8500	Logistic regression trained on synonyms, hyponyms and hypernyms on word2vec features with AUC maximization
9-ae-9	0.8342	Word2vec CBOW with window size 5 on Russian National Corpus, augmented with skip-gram model with context window size 20 on news corpus
9-ae-6	0.8268	Word2vec CBOW model with context window size 10 trained on web corpus
12-rt-3	0.5651	Applying knowledge extracted from Wikipedia and Wiktionary for computing semantic relatedness

Table 9. 10 best models according to the AE2 benchmark. Top 5 models are in bold font

Model ID	AE2-AVEP	Method Description
1-ae-1	**0.9849**	Desicion trees based on n-grams (Wikipedia titles and search queries), morphological features and Word2Vec
5-ae-3	**0.9835**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, bigrams on the same corpus, synonym database, prefix dictionary, orthographic similarity
5-rt-3	**0.9749**	Word2vec (skip-gram, window size 10, 300d vectors) on ruwac + lib.ru + ru-wiki, synonym database, prefix dictionary, orthographic similarity
1-rt-3	**0.9723**	Logistic regression trained on synonyms, hyponyms and hypernyms on word2vec features with AUC maximization
15-rt-1	**0.9669**	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 100d vectors, window size 10, 1 iteration, min cnt 100)
9-ae-6	0.9649	Word2vec CBOW model with context window size 10 trained on web corpus
15-rt-2	0.9646	Word2vec trained on 150G of texts from lib.rus.ec (skip-gram, 500d vectors, window size 5, 3 iteration, min cnt 5)
16-ae-1	0.9565	GloVe (100d vectors) on RuWac (lemmatized, normalized)
9-ae-9	0.9517	Word2vec CBOW with window size 5 on Russian National Corpus, augmented with skip-gram model with context window size 20 on news corpus
17-rt-1	0.9490	Distributional vector-based model, window size 5, trained on RUWAC and NRC, plmi-weighting
12-rt-3	0.7756	Applying knowledge extracted from Wikipedia and Wiktionary for computing semantic relatedness

СРАВНЕНИЕ ТРЕХ СИСТЕМ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ДЛЯ РУССКОГО ЯЗЫКА

Арефьев Н. В. (narefjev@cs.msu.su)
Московский Государственный Университет им. Ломоносова и ООО «Лаборатория Цифрового Общества», Москва, Россия

Панченко А. И. (panchenko@lt.informatik.tu-darmstadt.de)
Дармштадский Технический Университет, Дармштадт, Германия

Луканин А. В. (artyom.lukanin@gmail.com)
ООО «СофтПлюс», Челябинск, Россия

Лесота О. О. (cheesemaid@gmail.com)
Московский Государственный Университет им. Ломоносова, Москва, Россия

Романов П. В. (romanov4400@gmail.com)
ОАО 1С, Москва, Россия

Статья представляет результаты участия в дорожке по семантической близости RUSSE. Мы сравниваем три подхода к оценке семантической близости слов. Данные подходы основаны на использовании корпусов текстов русского языка. В первом подходе используются лексико-синтаксические шаблоны для извлечения и разметки предложений, содержащих слова, находящихся в гипо-гиперонимических отношениях. Второй подход — это классический метод контекстного окна на данных Google N-Grams. В третьем подходе используется программа *word2vec* и большой корпус для создания векторов слов. Последний метод считается лучшим методом для английского языка. Наши эксперименты показывают, что он также является лучшим методом для русского языка. В данной статье мы анализируем, как изменение метапараметров *word2vec* и использование различных корпусов, на которых он обучается, влияет на качество получаемых векторов слов. Мы также предлагаем простую, но действенную, методику по учёту слов, отсутствующих в словаре.

Ключевые слова: семантическая близость, лексико-синтаксические шаблоны, Google N-Grams, контекстное окно, word2vec, RUSSE, русский язык

EVALUATING THREE CORPUS-BASED SEMANTIC SIMILARITY SYSTEMS FOR RUSSIAN

Arefyev N. V. (narefjev@cs.msu.su)
Lomonosov Moscow State University &
Digital Society Laboratory, Moscow, Russia

Panchenko A. I. (panchenko@lt.informatik.tu-darmstadt.de)
TU Darmstadt, Darmstadt, Germany

Lukanin A. V. (artyom.lukanin@gmail.com)
LLC "SoftPlus", Chelyabinsk, Russia

Lesota O. O. (cheesemaid@gmail.com)
Lomonosov Moscow State University, Moscow, Russia

Romanov P. V. (romanov4400@gmail.com)
1C Company, Moscow, Russia

This paper reports results of our participation in the first shared task on Russian Semantic Similarity Evaluation (RUSSE). We compare three corpus-based systems that measure semantic similarity between words. The first one uses lexico-syntactic patterns to retrieve sentences indicating a particular semantic relation between words. The second one builds traditional context window approach on the top of Google N-Grams data to take advantage of the huge corpora it was collected on. The third system uses *word2vec* trained on a huge *lib.rus.ec* book collection. *word2vec* is one of the state-of-the-art methods for English. Our initial experiments showed that it yields the best results for Russian as well, comparing to other two systems considered in this paper. Therefore, we focus on study of *word2vec* meta-parameters and investigate how the training corpus affects quality of produced word vectors. Finally, we propose a simple but useful technique for dealing with out-of-vocabulary words.

Keywords: semantic similarity, lexico-syntactic patterns, skip-gram model, Google n-grams, context window, word2vec, RUSSE, Russian language

1. Introduction

A semantic similarity measure (SSM) outputs words with close meaning to an input word. For instance, such system can take as input the word "python" and return a list of related words, such as "perl", "ruby", "snake", "reptile" and "holy grail" (see serelex.org/#python). Similarity can be interpreted in many ways. In this paper, we consider words similar if they are synonyms, hypernyms or free (cognitive) associations, depending on the task. SSMs can be *global* and *contextual*. A global measure does not consider any context and therefore will return a mix of senses for ambiguous words, such as "python". On the other hand, contextualized SSMs take into account

context and therefore can filter out irrelevant results for a given word occurrence. Usually, a similarity measure returns a weighted (or ranked) list of results. However, most often, such a list contains a mix of synonyms, hyponyms, associations, co-hyponyms and other related words without explicit distinction between them.

The main motivation for development of SSMs is the wide range of language processing applications, they can be applied in, ranging from lexical substitution and word sense disambiguation to query expansion and question answering. No wonder many researchers tried to propose SSMs during the last two decades. In particular, most of the methods rely on a text corpus in order to estimate word similarities, for instance the classical distributional models, such as the context window and the syntactic context techniques. However, there exist many other original approaches that are built upon the structure of a lexical network, counts of a web search engine or entries of a dictionary. One of the recent trends in this field is corpus-based models that use a neural network to train word vectors used for similarity computation. The skip-gram model used in our work is one of them (Mikolov et al., 2013). You will be able to find exhaustive references to the mentioned above techniques in multiple comparisons of SSMs, such as Lee (1999), Agirre et al. (2009), Ferret (2010), Panchenko (2013) and Baroni (2014).

While there exist many approaches to semantic similarity, most of them were tested only for English. On the other hand, the Russian language has several important features that make it quite different from English: a grammar system with complex morphological rules, very flexible word order, absense of articles and Cyrillic alphabet. It is therefore premature to take for granted that the approaches yielding good results for English are going to work as well in the context of Russian. A recent paper by Zervanou et al. (2014) provides a study of semantic similarity for morphologically rich languages, including German and Greek, however Russian is not considered in the experiment. Finally, several researchers already tried to apply distributional semantic models for the Russian language including Krizhanovski (2007), Turdakov (2010), Krukov et al. (2010), Sokirko (2012) and Kolb[1]. However, these experiments lack a systematic evaluation of semantic similarity measures for Russian. Indeed, the workshop on Russian Semantic Similarity Evaluation *RUSSE* (Panchenko et al., 2015) introduced the first large-scale publicly available evaluation framework tailored for the Russian language. In this work, we use this collection of novel benchmarks to assess performance of our approaches[2].

Main contribution of our work is a comparative study of three global corpus-based systems of semantic similarity for the Russian language that are based respectively on the lexico-syntactic patterns, the right side context window and the skip-gram model. To the best of our knowledge, this is the first public attempt to quantify performance of these three approaches in the context of the Russian language. We experimentally assess performance of these techniques in the context of a shared task on a Russian semantic similarity, where the proposed methods consistently score in the top 10 models in all tracks. Systems and models described in this paper are available online (see below). In particular, to the best of our knowledge, we are the first to release a large scale *word2vec* model for the Russian language.

---

[1] http://www.linguatools.de/disco/disco_en.html

[2] https://github.com/nlpub/russe-evaluation/tree/master/russe/evaluation, http://russe.nlpub.ru

2. The First System: Pattern-Based Similarity on Wikipedia and Web corpora

The *PatternSim* similarity measure was first introduced for English language by Panchenko et al. (2012). The method operates in two steps. First, it extracts a set of sentences, which contain similar words, and tags these words with a set of manually crafted lexico-syntactic patterns. Second, it calculates a semantic similarity between words based on several factors, such as a term frequency and the number of term co-occurrences within sentences. Implementation of the method is available online[3].

2.1. Corpora

We used two corpora in order to calculate the semantic similarity with the *PatternSim* measure: the Russian Wikipedia and a collection of Russian Web pages. The Wikipedia dump was downloaded in April 2014 and processed with the *WikpediaExtractor.py* script[4]. The corpus of Web pages was crawled from the pages of 2,736 web sites each belonging to one of 20 following topical categories: auto-moto, beauty, child wares, clothes, clubs-concerts-cinema, cookery, credits, eating-out, everyday wares, furniture, insurance, information technology, massage, medicine, politics, realty, religion, repair wares, sport, travel. The seed web sites and the corpus itself are available for download[5].

Table 1. Corpora used by the three similarity measures described in this paper

Name	Description	Tokens	Documents	Size, Gb
wiki	Russian Wikipedia	238,052,379	1,159,723	3
web	Russian Web Pages	567,914,057	890,551	7
lib	Lib.rus.ec book collection	12,902,854,351	233,876	149
ngram	Russian Google N-Grams	67,137,666,353	591,310	—

2.2. Lexico-syntactic patterns

Sabirova and Lukanin (2014) developed six lexico-syntactic patterns for extracting hypernyms and hyponyms from Russian texts. The patterns were encoded as a cascade of finite state transducers (FSTs) with the help of the corpus processing tool *Unitex*[6]. Our grammar relies on the full version (Nagel, 2002) of the standard Russian morphological dictionary shipped with the tool. We apply these FSTs to mark hypernyms and hyponyms

[3] https://github.com/cental/PatternSim

[4] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[5] http://panchenko.me/data/dataset-2734.csv,
http://panchenko.me/data/webtopic-corpus-892233.csv.gz

[6] http://www-igm.univ-mlv.fr/~unitex/

with special tags (HYPER and HYPO). For example, the first FST of six, which corresponds to pattern "такие/таких/таким HYPER, как HYPO[, HYPO] и/или HYPO" (such HYPER as HYPO[, HYPO], and/or HYPO), will produce the following tagged sentence:

> В Индии зародились такие {[религии]=HYPER} как {[индуизм]=HYPO}, {[буддизм]=HYPO}, {[сикхизм]=HYPO} и {[джайнизм]=HYPO}.
>
> In India such {[religions]=HYPER} as {[Hinduism]=HYPO}, {[Buddhism]=HYPO}, {[Sikhism]=HYPO}, and {[Jainism]=HYPO} were born.

Such tagged sentences are used in order to estimate similarity between words. In this case, the words *religion*, *Hinduism, Buddhism, Sikhism,* and *Jainism* will be considered to be semantically similar (see the next section).

2.3. Calculation of semantic similarity

We experimented with different ranking formula and the metric *Efreq-Rnum-Cfreq-Pnum* proved to work best for English and French languages (Panchenko et al., 2012). This metric relies on several factors:

- the number of term co-occurrences within a set of concordances;
- frequencies of related terms;
- the "hubness" of related terms; the similarity with the terms that are related to many other terms is reduced;
- the number of distinct patterns which extracted a relation; relations extracted independently by several patterns are more robust than those extracted only by one pattern.

3. The Second System: Right-Context Window on the Google N-Grams

The right-context window distributional model represents each word as a vector in a vector-space built using Google N-grams corpus. Its dimensionality is equal to the number of unique words in the corpus, called contexts (or context words), thus each dimension is associated with exactly one context. In the model each element of a vector of any word contains information about its co-occurrence with a certain context word. Semantic similarity calculation is based on an assumption that two words similarity correlates with the distance between their vectors.

3.1. Corpus

The Google N-grams project aims at collecting statistical data of all ever published books, using Google Books corpus[7]. The Russian section of this corpus consists

[7] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html, https://books.google.com

of 591 thousand volumes and contains over 67 billion tokens. Every section includes a subsection per each type of N-grams, for N from 1 to 5. Each subsection presents information about N-grams passed certain occurrence thresholds. The full N-gram list is not publicly available. The information is formatted as such lines: "*NG*, *Y*, *P*, *V*, *C*", where *C* is the number of times N-gram *NG* appeared in the corpus in the year *Y*, while *P* and *V* display numbers of pages and volumes containing the N-gram respectively.

The main advantage of the Google N-gram corpus is its size. In our model, we use data from the year 1900 to the present time for preserving language integrity, which is about 560 thousand volumes and over 64 billion tokens. However, in this corpus, the information is sorted by the first word of N-gram, thus preventing the researchers from conducting an experiment for symmetrical context window in reasonable time, which is a more common approach (Patel et al., 1997). In addition it turned out that due to occurrence thresholds many words do not get enough contexts to represent their meanings.

3.2. Calculation of semantic similarity

In the experiment, semantic similarity between two words is modeled by a cosine distance between two corresponding PPMI (Positive Pointwise Mutual Information) (Bullinaria and Levy, 2007) vectors. Component a_i (corresponding to a context word cw_i) of a PPMI vector of a word w is calculated as follows:

$$a_i = max(0, log(\frac{P(w, cw_i)}{P(w) \cdot P(cw_i)})) = max(0, log(\frac{count(w, cw_i) \cdot count(.)}{count(w) \cdot count(cw_i)})),$$

where $P(w, cw_i)$ is the probability of the occurrence of cw_i within the distance of five words to the right of w (since right-context window of width 5 was used), $P(w)$ and $P(cw_i)$ are probabilities of words w and cw_i, correspondingly; $count(w, cw_i)$, $count(w)$, $count(cw_i)$ are corresponding frequencies and $count(.)$ is the size of the corpus.

4. The Third System: Skip-Gram Model on the LibRusEc corpus

word2vec is a piece of software developed by Mikolov et al. (2013) for learning vector representations for words and phrases[8]. These representations are learnt as the result of parameter optimization for a probabilistic language model. *word2vec* supports several language models. Here we will briefly describe just the one we used to obtain the best results, namely the *skip-gram model*, that was trained using the negative sampling method. Like the widely used bigram model, the skip-gram model estimates probability for a pair of words to be close to each other in the text. But unlike the bigram model these words do not have to occupy adjacent positions, instead they can be separated by other words.

[8] https://code.google.com/p/word2vec

Assume $P(D = 1|w, c; \theta)$ is the probability of the event that a word w appears in some context c. Then $P(D = 0|w, c; \theta) = 1 - P(D = 1|w, c; \theta)$ is the probability of the opposite event. Originally, the set of word's contexts is just the set of words occurring within some predefined distance (window size, *win*) from the target word w. However, the model generalizes to other context types; for instance, in (Levy and Goldberg, 2014) syntactically dependent words were used as contexts. We want the model to assign high probability to (c, w) pairs which can appear in texts and low probability to the ones which cannot. So the authors of the skip-gram model defined the following optimization problem:

$$\theta^* = arg\ max \prod_{(c,w)\in corp} P(D = 1|w, c; \theta) \prod_{(c,w)\in rand} (1 - P(D = 1|w, c; \theta))$$

Here *corp* contains (c, w) pairs extracted from corpus and *rand* contains randomly generated (c, w) pairs. The probability is calculated the following way:

$$P(D = 1|w, c; \theta) = (1 + e^{-V_c W_w})^{-1},$$

where $\theta = (V, W)$ are two matrices which columns V_c and W_w contain vectors of context c and the word w of some predefined length (vector dimensionality, *dim*). Thus optimization process gives us context vectors and word vectors. We ignore the former and use the latter to calculate semantic similarity. It was shown that simple algebraic operations with such word vectors can be used to model different semantic relations between corresponding words (Mikolov et al., 2013). For instance, synonyms will have very similar word vectors in the terms of cosine measure.

There exist several implementations of the skip-gram model. We used the original C implementation provided by the authors of the method to build word vectors and the Python implementation which is a part of the *GenSim* library[9] to calculate semantic similarity between words given their vectors.

4.1. Corpus

Lib.rus.ec is a large collection of Russian books in machine-readable XML-based format FB2. Each FB2 file contains meta information about a particular book (title, language, author, etc.) and its text. Using this meta information we selected books written in Russian. Texts of these books were saved as a single 149G text file containing 12.9 billion tokens[10].

Along with Lib.rus.ec we tried vectors trained on non-lemmatized and lemmatized versions of Russian Wikipedia (see Table 1) and also a version where each token was a concatenation of the lemma and the POS tag e.g. "*российский#JJ империя#NN*" (russian#JJ empire#NN).

---

[9] http://radimrehurek.com/gensim/

[10] as reported by the Unix command *wc*

4.2. Calculation of semantic similarity

Preprocessing. Each corpus, used to build word vectors, was preprocessed with a slightly modified script from *word2vec* C distribution. The script converts text to lowercase, inserts space character before punctuation marks (otherwise they are considered a part of the previous word), removes digits, several special characters, etc. We also added some preprocessing, that is specific for Russian (replaced all occurrences of "ё" to "е", for instance).

Building word vectors. To build word vectors an appropriate utility from *word2vec* C distribution was executed on the preprocessed corpus. We specified the following parameters:

- *cbow*: train CBOW (context bag of words) or skip-gram model. As our preliminary experiments showed, the skip-gram model always gives better results than CBOW, so we did not use CBOW for our submissions and do not describe it here.
- *dim*: word vectors dimensionality; we tried values from 100 to 1,000.
- *window*: maximum distance between a target word and words counted as its contexts; we tried values from 2 to 30.
- *iter*: number of passes over the whole corpus; to solve optimization problem described earlier, *word2vec* uses stochastic gradient descent—an iterative method which can benefit from processing the same training examples many times.
- *min-cnt:* discard words which appear less than this number of times in the corpus. We specified *min-cnt* = 5.

  All other parameters were not specified, so the default values were used.

Calculating distance. To calculate a semantic similarity between words we calculated cosine between the corresponding vectors. To deal with out-of-vocabulary words, i.e. the words which didn't occur in our corpus or occurred less than *min-cnt* times, we tried the following technique denoted as "*oov*" in the results table. If a vector is missed for one or both words from a particular word pair we used a set of vectors of its parts instead. First, we tried to split out-of-vocabulary words by a dash and for each in-vocabulary part added its vector to the set. If such set was still empty we tried to remove prefixes from such a word and if the derived words had vectors, then we added their vectors to the set. For instance, the word "*авиамотосообщение*"—a composite noun meaning flight or automobile connection—was represented with vectors of "*мотосообщение*" (automobile connection) and "*сообщение*" (transport connection). We defined similarity between two sets of vectors as similarity between the most similar vectors from these sets. The following examples illustrate the described technique:

sim(актриса, актер-статист) = sim(актриса, [актер, статист]) =
sim(актриса, актер) = 0.75
sim(автотехника, автомототехника) = sim(автотехника,
[мототехника, техника]) = sim(автотехника, мототехника) = 0.64

sim(actress, dummy-actor) = sim(actress, [dummy, actor]) = sim(actress,
actor) = 0.75
sim(auto-vehicles, auto-motor-vehicles) = sim(auto-vehicles,
[motor-vehicles, vehicles]) =
sim(auto-vehicles, motor-vehicles) = 0.64

5. Results and Discussion

We performed evaluation of the systems described above on the shared task on a Russian semantic similarity *RUSSE*. This shared task provided us with four benchmarks:

1. **HJ.** Correlations with human judgements in terms of Spearman's rank correlation. This test set was composed of 333 word pairs.
2. **RT**. The quality of a semantic relation classification in terms of an average precision. This test set was composed of 9,548 word pairs (4,774 unrelated pairs and 4,774 synonyms and hypernyms from the *RuThes Lite* thesaurus[11]).
3. **AE**. The quality of a semantic relation classification in terms of an average precision. This test set was composed of 1,952 word pairs (976 unrelated pairs and 976 cognitive associations from the Russian Associative Thesaurus[12]).
4. **AE2.** The quality of a semantic relation classification in terms of an average precision. This test set was composed of 3,002 word pairs (1,501 unrelated pairs and 1,501 cognitive associations from a web-scale associative experiment[13]).

Table 2 presents the results of the three methods on the shared task. As one can observe, the similarity measure *PatternSim* based on lexico-syntactic patterns yields the best results on the concatenation of Wikipedia and Web corpora. However, the *PatternSim* measures provide one of the lowest results among the three considered approaches in terms of correlations with human judgements (*HJ*). Average precision of this method on synonyms and hypernyms (*RT*) and free associations (*AE2*) is also rather low as compared to top system in our study and other best submission to the RUSSE shared task.

Table 2. Comparisons of the the HJ, RT, AE and AE2 datasets

Method	Corpus	HJ	RT	AE	AE2
patternsim	web+wiki	0.372	0.754	0.708	0.797
patternsim	wiki	0.322	0.755	0.724	0.784
patternsim	web	0.322	0.745	0.696	0.775
skipgram-dim100-win10-iter1	lib	0.621	0.847	0.912	**0.967**
skipgram-dim500-win20-iter1 + oov	**lib**	**0.677**	**0.905**	0.907	**0.965**
skipgram-dim300-win20-iter1	lib (20%)	0.651	0.856	**0.917**	**0.965**
skipgram-dim500-win5-iter3	lib	0.654	**0.903**	0.912	**0.965**
skipgram-dim500-win5-iter3	*wiki_nonlem.*	*0.532*	*0.731*	*0.881*	*0.914*
skipgram-dim500-win5-iter3	*wiki*	*0.601*	*0.803*	*0.771*	*0.928*
skipgram-sim500-win10-iter3	*lib*	*0.674*	*0.903*	*0.925*	*0.972*
skipgram-sim500-win10-iter3 + oov	***lib***	***0.699***	***0.918***	***0.928***	***0.975***
right-context-window	ngram	0.303	0.612	0.734	0.676

[11] http://www.labinform.ru/pub/ruthes/index.htm

[12] http://it-claim.ru/asis

[13] http://sociation.org

A more close inspection of the results of the pattern-based measures shows that a low performance is caused by a low recall of this approach. The method yields high precision, but is not able to assess similarity between some word pairs. Indeed, this model was able to assess similarity of 5–30% of word pairs, depending on the dataset. For instance, the method PatternSim on the web+wiki corpus was able to model only 98 of 333 word pairs. Therefore, sparsity of this representation is the main problem of the current version of this system.

According to our experiments, the right-context-window approach showed lowest scores among the three considered systems, despite the fact that Google N-gram corpus is 5 times bigger than the Lib.rus.ec. We think the main reason is the frequency threshold which ngrams must pass to be included in the corpus. We investigated occurrences of several less frequent words in Google N-gram corpus and found that there are too few contexts to build an adequate vector representations for these words. Probably, the threshold should not be constant, but should instead depend on the frequency of a particular word.

Finally, the skip-gram model yielded the best results according to the *RUSSE* evaluation. Even when trained on a non-lemmatized Wikipedia, it gives better results than the other two systems, except for the *RT* metric, where it performs almost the same as *PatternSim*. Training on a lemmatized Wikipedia improves the model even further. Finally, the model trained on non-lemmatized Lib.rus.ec showed even better results as this corpus is 50 times bigger than the Russian Wikipedia. It would be interesting to use a lemmatized and POS-tagged version of Lib.rus.ec but we leave this experiment for the future. Increasing corpus size gives significant improvements which are especially notable on the *RT* metric. In the shared task, our skip-gram system ranks among the top 10 submissions (out of 105 other systems), or in the top 5 participants (out of 19 other participants) according to all metrics[14]. The best skip-gram models for Russian language and scripts required to train and use them are available online[15].

To gain more insights on how *word2vec* meta-parameters influence performance, we evaluated models trained with different parameters and on different corpora. We display the most interesting results in Table 2, the full results table is available online[16]. In table 2 we also include the results which were not submitted because they were obtained after the submission was closed. These results are included for comparison and are displayed in *italics*. Several conclusions can be made from these results.

First of all, some preprocessing of the corpus is necessary, otherwise the results could be 3–12% worse than they could be. Probably this is because *word2vec* treates punctuation marks as a part of a previous word if they are not separated by a white space. The lemmatized version of Wikipedia gives about 10% improvement on *HJ* and *RT* metrics compared to the non-lemmatized version, however on *AE2* metric the improvement is only 3% and on *AE* metric the non-lemmatized version is 7% better. Probably this happens because an association word often agrees with a stimulus word in gender and number, so it is not lemmatized.

---

[14] http://russe.nlpub.ru/results

[15] https://github.com/nlpub/russe-evaluation/tree/master/russe/measures/word2vec

[16] http://goo.gl/xPL7DT

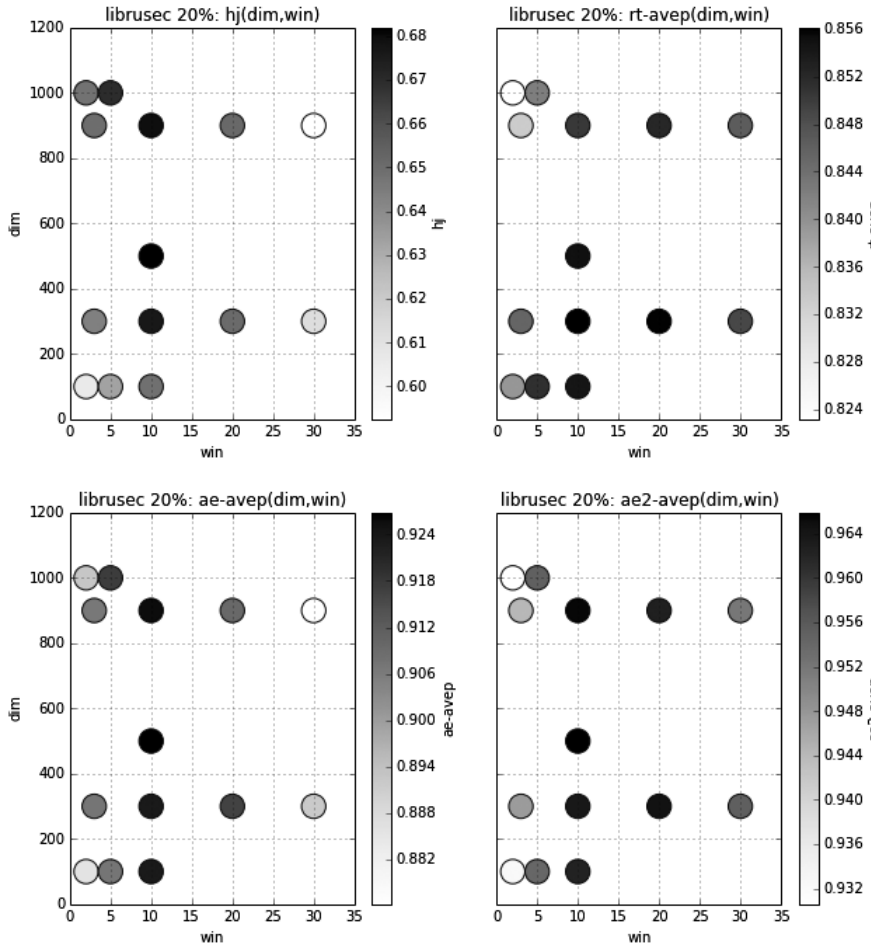


Fig. 1. Dependence of *word2vec* vectors' performance from the window size and vectors dimensionality. Vectors were trained on 20% of librusec corpus (30G)

We investigated how word vector dimensionality and context window size affect the results. We did it on 20% of Lib.rus.ec to be able to try many parameter combinations while reducing the computation time. However, it seems that on 100% librusec the results are similar. Fig. 1 clearly shows that performance declines when the window size is less than 5 or more than 20, window size 10 seems to be optimal among the window sizes we tried. The vectors dimensionality does not affect performance as much as the window size, dimensions between 300 and 900 give close results.

Fig. 2 shows how the results depend on the corpus and the number of iterations. As we said before, using even 20% of the non-lemmatized librusec (30G) instead of the lemmatized Wikipedia (3G) gives huge improvements (about 10% on hj and

rt metrics, 20% on *AE* and 3% on *AE2*). However using the whole librusec (150G) gives little improvement on *RT* and *AE2* metrics and even degradation on *HJ* and *AE* metrics compared to 20% of librusec. We have also compared the results on Wikipedia and 2% of librusec, which is almost the same size as Wikipedia (3G). The results on *HJ* and *AE2* are comparable with the lemmatized wiki, on *AE* 2% of librusec give better results which are comparable to the non-lemmatized wiki and on *RT* the lemmatized wiki beats both its non-lemmatized version and 2% of librusec with a huge gap.

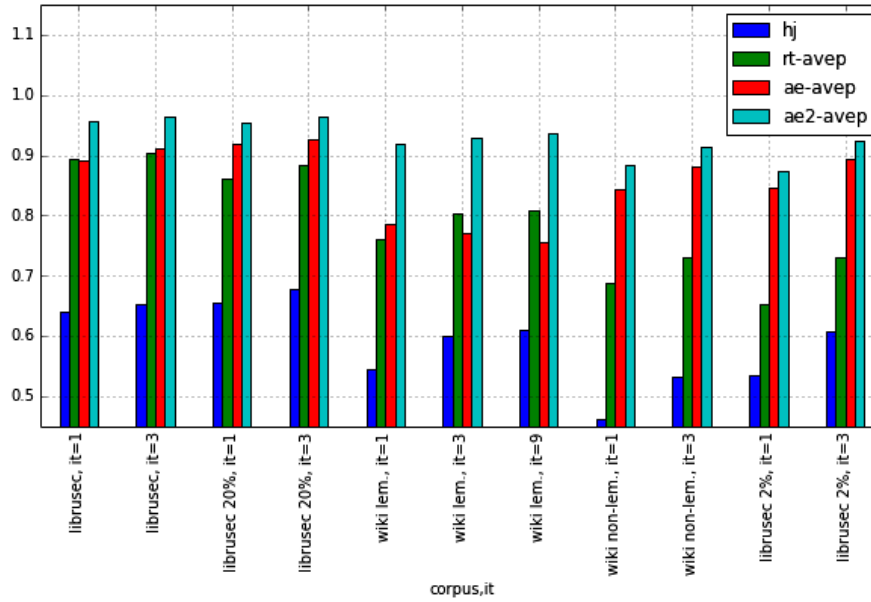


Fig. 2. Dependence of *word2vec* vectors' performance from the corpus and number of iterations. All vectors are 500d, window size is 5

We found that increasing the number of iterations over the whole corpus (*iter* parameter) gives great improvements on small corpora, such as Wikipedia, and little, but uniform improvements on large corpora; however the training time increases proportionally to the number of iterations, so it is very expensive to use large values of this parameter on large corpora. Finally, as one can see in Table 2, our technique for dealing with out-of-vocabulary words improves the results a little, but uniformly across all metrics.

6. Conclusion and Future Work

Our experiments clearly indicate that it is hard to compete with word vectors that are trained using *word2vec* and such a simple metric as the cosine distance between these vectors. Even when trained on a relatively small Russian Wikipedia this system performs better than the two other systems considered in this paper. When it is trained on larger corpora and good meta-parameters are selected it ranks

in the top 10 submissions (among other 105 submissions), or in the top 5 participants (among 19 other participants) according to all metrics of the *RUSSE* shared task. It worth to notice that these results were reached relatively easy by using freely available implementations of the *word2vec* method and small modifications of the preprocessing scripts to better handle Russian. Most time was spent on the selection of meta-parameters and corpora conversions. We also proposed a simple technique for dealing with compositional out-of-vocabulary words which gave a small but uniform improvement.

We showed that usage of the lemmatized version of Wikipedia instead of the non-lemmatized one gives better performing word vectors according to all metrics except one. We used a non-lemmatized version of Lib.rus.ec and leave experments with its lemmatization for the future. Another promising direction is training *word2vec* on Google N-Grams data which was collected on 5x larger corpora than Lib.rus.ec. However, usage of only Google N-Grams limits the window size to 2 (because only n-grams with n from 1 to 5 are available) which we found to be too small. So it is better to use a combination of Google N-Grams with other corpora. Two problems *word2vec* does not handle are words with multiple meanings and out-of-vocabulary words. These problems should be thoroughly considered in the future.

Acknowledgements

We thank Digital Society Laboratory LLC provided us with computational platform for most of the experiments described in this paper. Also we would like to acknowledge work of Kristina Sabirova developed the first version of the extraction patterns for *PatternSim* similarity measure.

References

1. *Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A.* (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In Proceedings of NAACL-HLT 2009, pages 19–27.
2. *Baroni, M., Dinu, G., & Kruszewski, G.* (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Vol. 1).
3. *Bullinaria, J., Levy, J.* (2007). Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study. In Behavior research methods 39 (3), pages 510–526.
4. *Van de Cruys, T.* (2010). Mining for Meaning: The Extraction of Lexicosemantic Knowledge from Text. PhD thesis, University of Groningen, The Netherlands.
5. *Curran, J. R.* (2004). From distributional to semantic similarity. PhD thesis. University of Edinburgh, UK.
6. *Ferret, O.* (2010). Testing semantic similarity measures for extracting synonyms from a corpus. In Proceeding of LREC.

7. *Krizhanovski A. A.* (2007), Evaluation experiments on related terms search in Wikipedia, SPIIRAS Proceedings, Vol. 5, pp. 113–116.
8. *Krukov K. V., Pankova L. A., Pronina V. S., Sukhoverov V. S., Shiplina L. B.* (2010), Semantic similarity measures in ontology, Control Sciences, Vol. 5, pp. 2–14.
9. *Lee, L.* (1999). Measures of distributional similarity. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, pages 25–32. Association for Computational Linguistics.
10. *Levy, O., Goldberg, Y.* (2014). Dependency-based word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Vol. 2, pp. 302–308).
11. *Mikolov, T., Chen, K., Corrado, G., Dean, J.* (2013). Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR.
12. *Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.* (2013). Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS.
13. *Nagel, S.* (2002). Formenbildung im Russischen. Formale Beschreibung und Automatisierung für das CISLEX-Wörterbuchsystem.
14. *Panchenko, A., Morozova, O., Naets, H.* (2012). A Semantic Similarity Measure Based on Lexico-Syntactic Patterns. In Conference on Natural Language Processing (KONVENS 2012), — Vienna (Austria), pp. 174–178.
15. *Panchenko, A.* (2013). Similarity measures for semantic relation extraction. PhD thesis. Université catholique de Louvain, 194 pages, Louvain-la-Neuve, Belgium.
16. *Panchenko A., Loukachevitch N. V., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015) "RUSSE: The First Workshop on Russian Semantic Similarity". In Proceeding of the Dialogue 2015 conference. Moscow, Russia
17. *Patel, M., Bullinaria, J. A., Levy, J. P.* (1997). Extracting Semantic Representations from Large Text Corpora. 4th Neural Computation and Psychology Workshop, London, 9–11 April 1997, 199–212.
18. *Sabirova, K., Lukanin, A.* (2014). Automatic Extraction of Hypernyms and Hyponyms from Russian Texts. In Supplementary Proceedings of the 3rd International Conference on Analysis of Images, Social Networks and Texts (AIST 2014) / Ed. by D. I. Ignatov, M. Y. Khachay, A. Panchenko, N. Konstantinova, R. Yavorsky, D. Ustalov. Vol. 1197: Supplementary Proceedings of AIST 2014. CEUR-WS.org, 2014. Pp. 35–40.
19. *Sahlgren, M.* (2006). The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. PhD thesis.
20. *Sokirko A.* (2013), Mining semantically similar language expressions for the Yandex information retrieval system (through to 2012) [Mayning blizkikh po smyslu vyrazheniy dlya poiskovoy sistemy Yandex (do 2012 goda)], available at: http://www.aot.ru/docs/MiningQueryExpan.pdf
21. *Turdakov D. Y.* (2010), Methods and software for term sense disambiguation based on document networks [Metody i programmnye sredstva razresheniya leksicheskoy mnogoznachnosti terminov na osnove setey dokumentov], PhD thesis, Lomonosov Moscow State University, Moscow, Russia.
22. *Zervanou, K., Iosif, E., Potamianos, A.* (2014). Word Semantic Similarity for Morphologically Rich Languages. LREC

ИСПОЛЬЗОВАНИЕ ФОЛКСОНОМИИ ДЛЯ ОПРЕДЕЛЕНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ

Клячко Е. (elenaklyachko@gmail.com)

Москва, Россия

Ключевые слова: семантическая близость, фолксономия, совместная категоризация, социальные сети

USING FOLKSONOMY DATA FOR DETERMINING SEMANTIC SIMILARITY

Klyachko E. (elenaklyachko@gmail.com)

Moscow, Russia

This paper presents a method for measuring semantic similarity. Semantic similarity measures are important for various semantics-oriented natural language processing tasks, such as Textual Entailment or Word Sense Disambiguation. In the paper, a folksonomy graph is used to determine the relatedness of two words. The construction of a folksonomy from a collaborative photo tagging resource is described. The problems which occur during the process are analyzed and solutions are proposed. The structure of the folksonomy is also analyzed. It turns out to be a social network graph. Graph features, such as the path length, or the Jaccard similarity coefficient, are the input parameters for a machine learning classifying algorithm. The comparative importance of the parameters is evaluated. Finally, the method was evaluated in the RUSSE evaluation campaign. The results are lower than most results for distribution-based vector models. However, the model itself is cheaper to build. The failures of the models are analyzed and possible improvements are suggested.

Keywords: semantic similarity, folksonomy, collaborative tagging, social networks

1. Introduction

Measuring semantic similarity is important for various natural language processing tasks, including Textual Entailment, Word Sense Disambiguation etc [1]. The aim of The First International Workshop on Russian Semantic Similarity Evaluation (RUSSE) [14] was to carry out an evaluation campaign of currently available methods for the Russian language.

The organizers provided several training sets. They also performed the evaluation on the test set.

2. Related work

2.1. Semantic similarity measurements

As described in [1], the approaches to semantic similarity measurement can be divided into knowledge-based ones or context-based ones. Knowledge-based approaches use taxonomies with pre-annotated world-relations. These taxonomies may be leveraged through collaborative tagging, for example:

1. tags made by software programmers for their projects at the FreeCode resource [18]
2. geographical tags at the Open Street Map project [3]
3. Flickr[1] image tags [16]
4. Del.icio.us[2] tags [16]

We can roughly divide the approaches to processing taxonomy data in the following groups. Naturally, features from different groups can be used jointly.

1. graph-based methods: the ontology is considered to be a graph
   a. in [1], a version of Page Rank is computed for both words, resulting in a probability distribution over the graph. Then the probability vectors are compared using cosine similarity measure
   b. in [4], path length features are used
2. ontology-based methods: these methods take into account the hierarchical structure of an ontology:
   a. in [4], the ratio of common and non-common superconcepts is calculated
   b. in [5], a feature which is based on the depth of the concepts and their least common superconcept is calculated
3. vector-space models: vectors are constructed, and their similarity is measured
   a. in [3], the vector space coordinates are words from term definitions, which were created as a part of a collaborative project.
   b. in [18], vectors of tf and idf scores are constructed. In [16], these vectors also have a temporal dimension

[1] https://www.flickr.com/

[2] https://delicious.com/

2.2. Pre-processing tags and refining tag structure

In [17], pre-processing techniques for folksonomy tags are described. These techniques involve normalizations and help cluster the tags better. In [12], the authors leverage user information in order to get a more precise understanding of tag meanings.

In [8], [10], and [15], a folksonomy is used for getting synonym and homonym relations between words. The authors reduce the dimensionality of the tag space by clustering the tags. Various measures are used, such as the Jaccard similarity coefficient, a mutual reinforcement measure, and the Jensen-Shannon divergence

In [2], lexico-syntactic patterns, which are traditionally used to get a taxonomy structure out of texts, are used to refine the taxonomy structure, which is constructed via obtaining tags from a collaborative resource.

2.3. Natural language generation

In a number of works, folksonomy structure is used in natural language generation tasks, namely for referring expression generation or text summarization [6, 13]

3. The goals of this paper

The aim of this work was to assess the contribution a folksonomy can make to word similarity measurements.

Vector-space models seem to be quite efficient for the word similarity task. However, such approaches are sometimes not easy to interpret linguistically, and using an ontology is sometimes preferrable. On the other hand, the construction of a manually-crafted ontology can take a lot of time. As a result, using a folksonomy seems to be an appropriate trade-off. The influence of various parameters of the folksonomy should also be investigated. Finally, studying the structure of a tag-based folksonomy as a quasi-natural object is quite interesting.

4. Folksonomy construction

For the RUSSE shared task, a folksonomy graph was built as a co-occurrence network of photo tags from Flickr.

The Flickr API was used to collect tags from photos in a database. The process was organized as follows:

1. start with an array of about 90,000 words (A. Zaliznyak's dictionary [19], the electronic version provided by SpeakRus[3]) and an empty graph.
2. for each *word1* in the array:
   a. get all photos tagged with *word1*

[3] http://speakrus.narod.ru/dict-mirror/

b. for each photo collected in (a):
   i. collect all other Russian-language tags from the photo. Use the number of photos to calculate the tag frequencies. As a result we get a number of (*word, frequency*) pairs.
   ii. for each *word2* with frequency *freq* (from the pairs collected in (i)) we create an edge in the graph: (*word1, word2, freq*)

Tables 1 and 2 show two fragments of the resulting co-occurrence matrix for the words "автобус" ('bus') and "ягода" ('berry'):

Table 1. A fragment of the frequency matrix for "автобус" ('bus')

word1	word2	word2 translation	frequency
автобус	природа	nature	146
автобус	улица	street	135
автобус	транспорт	transport	132
автобус	социалистически	socialist (*in Bulgarian language*)	91
автобус	комунистически	communist (*in Bulgarian language*)	90
автобус	россия	Russia	63
автобус	город	city	46
автобус	москва	Moscow	40
автобус	путешествия	travelling	40
автобус	корабль	ship	35

Table 2 A fragment of the frequency matrix for "ягода" ('berry')

word1	word2	word2 translation	frequency
ягода	россия	Russia	45
ягода	лето	summer	31
ягода	природа	nature	31
ягода	ягоды	berries	29
ягода	клубника	strawberry	28
ягода	красный	red	21
ягода	подмосковье	Moscow region	19
ягода	малина	raspberry	17
ягода	смородина	currant	16
ягода	еда	food	15
ягода	осень	autumn	15
ягода	флора	flora	15
ягода	москва	Moscow	14
ягода	вишня	cherry	13
ягода	дача	country cottage	13
ягода	черника	bilberry	13
ягода	дерево	tree	12

Language detection was the main issue at that stage. Flickr does not distinguish between the languages of the tags. The tags are also too short for a language detection tool to detect the language well enough. The Python-ported Google's detection library[4] was used for language detection. However, it soon turned out to filter some Russian words. As a result, Zaliznyak's dictionary itself was used as a source of additional checks. Probably, using a large corpus of Russian words would be a better way of detecting Russian-language words in this case. The publicly available data on the author of the tag could also be used.

The program to collect the data is a Python script available at https://github.com/gisly/word_similarity.

5. The resulting structure of the folksonomy

5.1. The folksonomy graph

The resulting folksonomy is a graph of 96,015 nodes and 1,015,992 edges. The mean node degree is approximately 21.16.

Logically speaking, the graph should be undirected because the co-occurrence relation should be symmetric. However, two problems made this impossible:

- the language detection bug described above led to the fact that sometimes *word1, word2* edge was present, but *word2, word1* was not because *word1* was not detected to be a Russian word
- the Flickr database is a not a snapshot: it is a continuously changing dataset. It means the same edge inconsistence as described above.

Naturally, the graph could have been made undirected after completing the download. However, we chose to leave it as it is and simply count for the edges' being directed.

5.2. The folksonomy graph as a complex network

What is interesting, the folksonomy graph turns out to be a complex network (in the same sense as a graph of people relations or a word co-occurrence graph; cf. [11]).

The node degree distributions fits the power-law, which is typical for a social network [11]. Fitting the power law[5], we a get a p-value of 0.99 for, which indicates the hypothesis of the power-law distribution cannot be rejected. The exponent value is 1.64.

The log node degree distribution graph is shown in fig. 1.

---

[4] https://pypi.python.org/pypi/langdetect

[5] the fit was made using the R package: http://www.inside-r.org/packages/cran/igraph/docs/power.law.fit

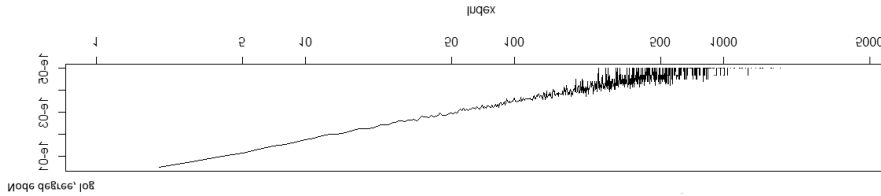


Fig. 1 Node degree distribution (log coordinates)

In table 3, top-10 words ordered by their degree are shown:

Table 3. Top-10 nodes ordered by node degree

word	translation	node degree
россия	Russia	4799
природа	nature	4096
красный	red	3875
москва	Moscow	3618
улица	street	3579
синий	blue	3543
солнце	Sun	3514
белый	white	3475
портрет	portrait	3366
отражение	reflection	3336

6. Training data

The RUSSE campaign consisted of two tasks. In the relatedness task, word relations (synonymy, hypo/hyperonymy were considered). In the association task, free associations were considered. As a part of the RUSSE evaluation campaign, several training and test datasets for each task were created by the organizers. The datasets are different in their origin. Some of were created through an online collaborative procedure, whereas others are extracted from large thesauri. A detailed description of these datasets as well as download links are given at the RUSSE website[6].

At first, these datasets contained only positive examples[7]. Therefore, we used a set of manually crafted negative examples. The negative examples were created by picking two random words from a large word set (the Wikipedia dump scores[8]), and manually excluding those which were really semantically similar to each other.

---

[6] http://russe.nlpub.ru/task/

[7] automatically generated negative examples were provided later

[8] https://s3-eu-west-1.amazonaws.com/dsl-research/wiki/wiki-cooccur-ge2.csv.bz2

During training, we mainly used the ***ae*** and ***rt*** training data, experimenting with different sizes of their subsets. ***ae*** are word association measures extracted based on an association. ***rt*** are word relatedness measures extracted from a thesaurus.

7. Features

For two words (*word1* and *word2*) the following features were calculated:

1. the existence of *word1* and *word2* nodes in the network (Y/N)
2. do *word1* and *word2* have the same part of speech[9]? (Y/N)
3. the existence of a path between *word1* node and *word2* node (Y/N)
4. path length: the number of nodes in the shortest path if the path exists (a number or NONE)
5. weighted path length (if the path exists; a number or NONE). In the shortest path, for each pair of nodes, the frequency of their joint occurrence is calculated. It is then divided by the frequencies of the individual words. The resulting measures are multiplied. Finally, a logarithm of the resulting number is taken.
6. the frequencies of the nodes in the path if the path exists (numbers or NONE). Each frequency is a separate feature.
7. the node degrees of the nodes in the path if the path exists (numbers or NONE). The degree of a node is the number of edges directly connected to the given node. Each degree is a separate feature.
8. the PageRank of the nodes in the path if the path exists (numbers or NONE)
9. the Jaccard similarity of *word1* node and *word2* node (a number). The Jaccard similarity coefficient is defined as:

   (the number of common neighbors of *word1* and *word2*)/(the size of the union of all neighbors of *word1* and *word2*)

10. the Dice similarity of *word1* node and *word2* node (a number). The Dice similarity coefficient is quite similar to the Jaccard coefficient and is defined as:

    2*(the number of common neighbors of *word1* and *word2*)/(the number of all neighbors of *word1* and *word2*)

11. the cosine similarity of the neighbor vector of *word1* and the neighbor vector of *word2*

[9] https://pythonhosted.org/pymorphy/ was used

7.1. The classification task

The classifiers were to solve the following task: each pair of words (*word1* and *word2*) should be classified as "similar" or "non-similar". Depending on the nature of the classifier, it was to produce either a binary score (0 or 1), or a number in the interval [0; 1]. In the latter case, the score was converted into the corresponding binary score:

- values ≤ 0.5 were considered to be 0
- values > 0.5 were considered to be 1

8. Machine learning algorithms

I tried several machine learning algorithms, such as Conditional Tree Inference, and Ada-Boost, implemented in the corresponding R packages (ctree[10] and ada[11]). The choice of these algorithms is mainly due to the fact that their results can be easier interpreted than the results of other algorithms.

8.1. Conditional Tree inference

A conditional tree is a kind of a decision tree. When building the conditional decision tree, the algorithm tests whether the hypothesis of the target variables's independence of the parameters can be rejected or not. If the hypothesis is rejected, it chooses the "strongest" parameter as a new node in the tree and proceeds with the other parameters [9].

In fig 2, the conditional tree which was built using the ***ae*** and ***rt*** subsets of the training data is presented.

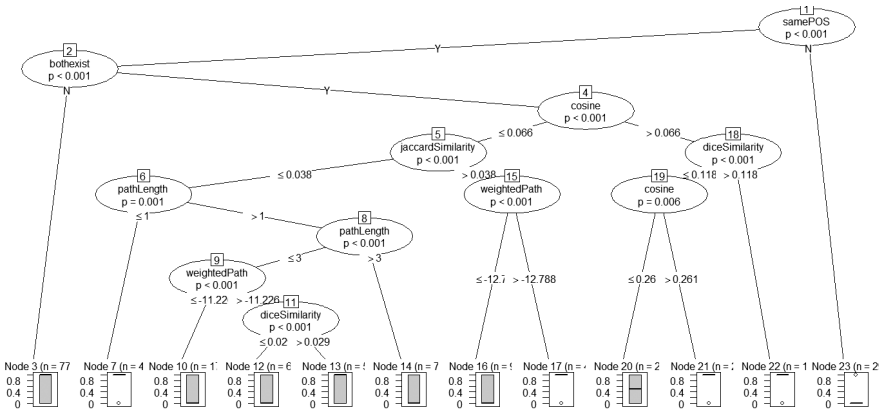


Fig. 2. The conditional tree created using the folksonomy graph and the ***ae*** training data subset

[10] http://cran.r-project.org/web/packages/partykit/vignettes/ctree.pdf

[11] http://cran.r-project.org/web/packages/ada/ada.pdf

8.2. AdaBoost

AdaBoost uses a committee of several weak classifiers (e. g., decision trees) and ends up calculating weights for these classifiers [7].

In fig 3, the variable importance plot constructed by AdaBoost is presented. The variable score shows the relative score of the variable.

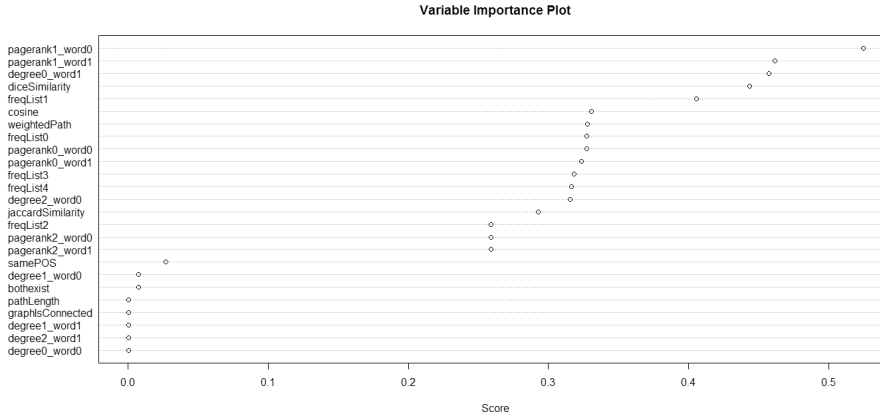


Fig. 3. The variable importance plot created by AdaBoost using the folksonomy graph and the ***ae*** training data subset

9. Evaluation

9.1. Cross-validation on the training set

I performed 4-fold cross-validation on the ae training set. The best average accuracy was 0.76 for the conditional tree model and 0.75 for the ada boost model. The best average precision was 0.73 for the conditional tree model and 0.70 for the ada boost model.

9.2. Final evaluation on the test set

Final evaluation was performed by the organizers[12]. The results for the folksonomy model are given in table[13] (model ids starting with "2-"):

---

[12] https://github.com/nlpub/russe-evaluation/tree/master/russe/evaluation

[13] from https://docs.google.com/spreadsheets/d/190qw6O_r8xAxPM2SK8q-R-0ODp2wDx-8qzh9Lr31jmSY/edit?usp=sharing

Table 4. The evaluation results for the folksonomy model provided by the organizers

HJ (human judgement for relatedness)	**RT-AVEP/ ACC** (average precision/ accuracy for ae-relatedness)	**AE-AVEP/ ACC** (average precision/ accuracy for ae associations)	**AE2-AVEP/ ACCURAC** (average precision/ accuracy for ae2 associations)	**Method Description**
0.3717	0.6815/0.5670	0.5195/0.4652	0.7282/0.6369	ctree, larger training subset
0.2490	0.7275/0.5396	0.5985/0.4795	0.7301/0.5903	AdaBoost, smaller training subset
0.2436	0.7183/0.5354	0.5802/0.5194	0.6732/0.5550	AdaBoost, larger training subset

10. Analysis

10.1. Intrinsic analysis: variable importance

From the output of AdaBoost and ctree, we can see that both algorithms consider the following parameters important:

- cosine similarity
- dice similarity
- jaccard similarity
- weighted path

Because of the structure of the network, the existence of the path itself does not mean much. Firstly, as we saw above, hubs such as “Russia”, “Moscow”, or “portrait”, which actually hold meta-information about a photo, connect most nodes with each other. Secondly, there may be an accidental connection between two words. For example, there is a photo tagged with words “egg” and “world” and it is an art representation of the world map on the eggshell. Naturally, it is an art concept and not the common truth.

Therefore, we should avoid two long paths because they may have a hub node inside. Moreover, we should avoid “accidental” paths.

The path length parameter and the weighted path parameter were thought to be the solution.

Actually, this intuition corresponds well enough with the ctree result: the larger the weighted path logarithm is, the greater is the probability of words being connected. It means the words are more likely to be related if the weighted path value is closer to one. Therefore, if the words are too frequent, we avoid considering them connected.

The conditional tree model also has two more important parameters: "both exist" and "same POS".

The scarcity of the photo tag data means that a lot of words simply lack. Therefore, the "both exist" feature simply prevents such words from being considered. However, naturally, the absence of the word in the folksonomy dictionary may only correlate with the word frequency in the everyday usage and not with its possible similarity with other words. For example, we cannot expect a folksonomy to have words like "яйцепродукты" ('egg products', a very special term from the food industry). Therefore, the parameter is perhaps useless and makes more noise than helps.

As regards the same POS feature, it is quite useful for the relatedness task because the common part of speech is usually considered to be important in the definitions of synonymy, hyponymy etc. However, it is really useless for the relatedness task.

There is also one intuitive problem with the ctree rules. According to them, if the similarity parameters are very low, but there is a direct link between the words, the words are considered to be related. In this case, the word frequencies are not analyzed at all.

10.2. Evaluation results

The algorithm performed quite consistently with the cross-validation results and considerably worse than the other competing methods.

10.2.1. Test set variations

We could expect that the photo tag similarity means association closeness and not relatedness. Moreover, we chose more ***ae*** training data as a training set. Therefore, the method was expected to work better on the association task than on the relatedness task.

Actually, the method does perform best on the ***ae2*** test set, which is a result of an online association experiment. The main reason for the poor performance on the Russian Associative Thesaurus test set is the absence of the thesaurus words in the folksonomy dictionary.

As regards the relatedness task, the method performs quite well on the RuThes relatedness subset. However, the ***hj*** (human judgment) results are poor. Why is it so that the two subsets expose different behavior?

Firstly, a subset of ***rt*** data was used for training. Secondly, in ***hj*** a finer-grained similarity score is given to word pairs, which is harder to reproduce.

10.2.2. The problems and possible solutions

In the table below, we collected several typical cases of the model's and failures. We then speculate of the possible ways of improving the model. We also mention the model's successes to show that they are not accidental.

Table 5. Error analysis

word0	word1	predicted ctree	actual	explanation	the source of the problem
true positives					
бас ('bass')	звук ('sound')	0.96	1	large cosine and dice similarities	no problem
рис ('rice')	крупа ('groats')	0.70	1	small weighted path (through the word "традиционный" 'traditional'); the photos actually depict some traditional meals.	no problem
false positives					
армия ('army')	река ('river')	0.96	0	large cosine and dice similarities	the common neighbors are mostly names of places
каланча ('watchtower/a tall person')	павильон ('pavilion')	0.96	0	large cosine similarity	perhaps, it is an annotation error. In my opinion, it is doubtful that the words are not co-hyponyms
сварщик ('welder')	изолятор ('isolation ward')	0.70	0	small weighted path through the words "синий" 'blue' and "Ярославль" 'Yaroslavl, a name of a city'	the words in the path are a hub and a name of a place
введение ('introduction')	сингл ('single (song)')	0.74	0	the word "single" does not exist in the database, so the "both exist" parameter works	the scarcity of the words and the "both exist" parameter
диагональ ('diagonal')	самолет ('airplane')	0.78	0	there is a direct link between the words.	The similarity parameters (cosine, dice etc) are very low, and in this case the path length parameter works. Perhaps, more careful analysis of negative examples could prevent such a rule from appearing in the ctree. However, such accidental links are intrinsic to the folksonomy so little can be done about it in general.
false negatives					
крыша ('roof')	ветхая ('dilapidated')	0.01	1	the word "ветхая" is not present among the tags in the given form. Furthermore, they have got different part of speech categories.	The corresponding masculine form of the adjective is present. The word can also be found in the photo descriptions
неделя ('week')	зачетная ('of exams'; the whole means 'exam week')	0.01	1	the words have different part of speech categories. Moreover, they are not connected well enough	the words co-occur in the photo descriptions but not in tags. The POS parameter is perhaps harmful

In order to improve the results, the following should be considered:

1. the hubs and place names usually contain meta-information, and do not depict the object shown in the photo. They should be filtered or somehow penalized. It can be done using geography databases and the graph statistics
2. all forms of a word should be considered. It can be achieved with a morphological analyzer.
3. photo descriptions and comments to photos should also be considered. They are accessible via the Flickr API.
4. more tags can actually be downloaded using more seed data, and adding non-vocabulary data
5. better language detection can be done (e. g., using a larger word list or simply taking all Cyrillic letter words)

10.3. Overall contribution

Although collecting the tags was inspired by the RUSSE shared task, the work has independent results, too. The way the folksonomy has been collected turns out to be valid because the resulting structure can be easily interpreted. Therefore, the method presented can be used in other natural language processing tasks (e. g., natural language generation, recommending services). Moreover, as far as we know, there are no similar publically shared open folksonomies for the Russian language

However, the problems we faced show that the data is very noisy and that we should pay more attention to normalizing it. Firstly, we should have paid more attention to the language detection problem. Secondly, the origin of the data should have taken into account. As the tags are connected with photos, they contain a lot of extra-linguistic information, which should be dealt with.

References

1. *Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A.* (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In Proceedings of NAACL-HLT 2009, pages 19–27.
2. *Almoqhim, Fahad, David E. Millard, and Nigel Shadbolt* (2014) "Improving on Popularity as a Proxy for Generality When Building Tag Hierarchies from Folksonomies." Social Informatics. Springer International Publishing. 95–111.
3. *Ballatore, Andrea, David C. Wilson, and Michela Bertolotto.* (2013) "Computing the semantic similarity of geographic terms using volunteered lexical definitions." International Journal of Geographical Information Science 27.10: 2099–2118.
4. *Banea, Carmen,* et al. (2012) "Unt: A supervised synergistic approach to semantic text similarity." Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. Association for Computational Linguistics.

5. *Batet, Montserrat, et al.* (2013) “Semantic similarity estimation from multiple ontologies.” Applied intelligence 38.1: 29–44.
6. *Boydell, Oisin, and Barry Smyth* (2007) “From social bookmarking to social summarization: an experiment in community-based summary generation.” Proceedings of the 12th international conference on Intelligent user interfaces. ACM.
7. *Culp, Mark, Kjell Johnson, and George Michailidis* (2006) “ada: An r package for stochastic boosting.” Journal of Statistical Software 17.2: 9.
8. *Eynard, Davide, Luca Mazzola, and Antonina Dattolo* (2013) “Exploiting tag similarities to discover synonyms and homonyms in folksonomies.” Software: Practice and Experience 43.12: 1437–1457.
9. *Hothorn T.* et al. (2011) Package ‘party’: A laboratory for recursive partitioning.
10. *Mousselly-Sergieh, Hatem, et al.* (2013) “Tag Similarity in Folksonomies.” INFORSID.
11. *Newman, Mark* (2008) “The physics of networks.” Physics Today 61.11: 33–38.
12. *Niebler, Thomas,* et al. (2013) “How tagging pragmatics influence tag sense discovery in social annotation systems.” Advances in Information Retrieval. Springer Berlin Heidelberg. 86–97.
13. *Pacheco, Fabián, Pablo Ariel Duboue, and Martín Ariel Domínguez.* (2012) “On the feasibility of open domain referring expression generation using large scale folksonomies.” Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies. Association for Computational Linguistics.
14. *Panchenko A., Loukachevitch N. V., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015) “RUSSE: The First Workshop on Russian Semantic Similarity”. In Proceeding of the Dialogue 2015 conference. Moscow, Russia
15. *Quattrone, Giovanni,* et al. (2012) “Measuring similarity in large-scale folksonomies.” arXiv preprint arXiv:1207.6037.
16. *Radinsky, Kira,* et al. (2011) “A word at a time: computing word relatedness using temporal semantic analysis.” Proceedings of the 20th international conference on World wide web. ACM.
17. *Solskinnsbakk, Geir, and Jon Atle Gulla.* (2011) “Mining tag similarity in folksonomies.” Proceedings of the 3rd international workshop on Search and mining user-generated contents. ACM.
18. *Wang, Shaowei, David Lo, and Lingxiao Jiang.* (2012) “Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging.” Software Maintenance (ICSM), 2012 28th IEEE International Conference on. IEEE.
19. *Zaliznyak, A.* (1977). Grammaticeskij Slovar’ Russkogo Jazyka. Slovoizmenenie (‘The Grammatical Dictionary of the Russian Language. Inflection’), Moscow: Russkij Jazyk.

НА ВХОДЕ ТЕКСТЫ, НА ВЫХОДЕ СМЫСЛ: НЕЙРОННЫЕ ЯЗЫКОВЫЕ МОДЕЛИ ДЛЯ ЗАДАЧ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ (НА МАТЕРИАЛЕ РУССКОГО ЯЗЫКА)

Кутузов А. (akutuzov@hse.ru)
НИУ Высшая Школа Экономики
и Mail.ru Group, Москва, Россия

Андреев И. (i.andreev@corp.mail.ru)
Mail.ru Group, Москва, Россия

Ключевые слова: искусственные нейронные сети, машинное обучение, семантическая близость, дистрибутивная семантика, векторные репрезентации лексики, word2vec

TEXTS IN, MEANING OUT: NEURAL LANGUAGE MODELS IN SEMANTIC SIMILARITY TASKS FOR RUSSIAN

Kutuzov A. (akutuzov@hse.ru)
National Research University Higher School of Economics and
Mail.ru Group, Moscow, Russia

Andreev I. (i.andreev@corp.mail.ru)
Mail.ru Group, Moscow, Russia

Distributed vector representations for natural language vocabulary get a lot of attention in contemporary computational linguistics. This paper summarizes the experience of applying neural network language models to the task of calculating semantic similarity for Russian. The experiments were performed in the course of Russian Semantic Similarity Evaluation track, where our models took from 2nd to 5th position, depending on the task.

We introduce the tools and corpora used, comment on the nature of the evaluation track and describe the achieved results. It was found out that Continuous Skip-gram and Continuous Bag-of-words models, previously successfully applied to English material, can be used for semantic modeling of Russian as well. Moreover, we show that texts in Russian National Corpus (RNC) provide an excellent training material for such models, outperforming other, much larger corpora. It is especially true for semantic relatedness tasks (although stacking models trained on larger corpora on top of RNC models improves performance even more).

High-quality semantic vectors learned in such a way can be used in a variety of linguistic tasks and promise an exciting field for further study.

Keywords: neural embeddings, machine learning, semantic similarity, distributional semantics, vector word representations, word2vec

1. Introduction

This paper describes authors' experience with participating in Russian Semantic Similarity Evaluation (RUSSE) track. Our system was trained using neural network language models; the process is explained below, together with the workflow for evaluation. We also comment on the nature of the RUSSE tasks and discuss features of neural models for Russian.

Since Ferdinand de Saussure, it is known that linguistic sign (including word) is arbitrary. It means that there is no direct connection between its form and concept (meaning). Consequently, printed orthographic words *per se* do not contain sense. What is important for the task discussed here, is that if given only disjoint word forms, a computer (an artificial intelligence) can't hope to grasp the concepts behind them and decide whether they are semantically similar or not.

At the same time, detecting degree of semantic similarity between lexical units is an important task in computational linguistics. The reason is threefold. First, it is a means in itself: often, applications demand calculating the "semantic distance" between words, for example, in finding synonyms or near-synonyms for search query expansion or other needs [Turney and Pantel 2010]. Second, once we know which words are similar and to what extent, we can "draw a semantic map" of the language in question and use this knowledge in a multitude of tasks, from machine translation [Mikolov et al. 2013b] to natural language generation [Dinu and Baroni 2014]. Finally, measuring performance in semantic similarity task is a convenient way to estimate soundness of a semantic model in general.

Consequently, various methods of overcoming linguistic arbitrariness and calculating semantic similarity for natural language texts were invented and evaluated for many widespread languages. However, computational linguistics community lacks experience in computing semantic similarity for Russian texts. Thus, the task of applying state-of-the-art methods to this material promised to be interesting, and kept its promise.

The paper is structured as follows. In the Section 2 we give a brief outline of RUSSE evaluation track. The Section 3 describes the models we used to compute semantic similarity and the corpora to train these models on. In the Section 4, results are evaluated and influence of various model settings discussed. The Section 5 lists the main results of our research. In the Section 6, we conclude and propose directions for future work.

2. Task Description

RUSSE[1] is the first attempt at semantic similarity evaluation contest for Russian language. It consists of four tracks: two for the relatedness task and two for the association task. Participants were presented with a list of word pairs and had to fill in the degree of semantic similarity between each pair, in the range [0;1].

[1] http://russe.nlpub.ru; the authors of the present paper are under the number 9 in the participants' list.

In the semantic relatedness task, participants were to detect word pairs in synonymic, hyponymic or hypernymic relations and to separate them from unrelated pairs. First track test set in this task included word pairs with human-annotated similarities between them. Systems' performance was measured with Spearman's rank correlation between these human scores and the system scores. The second track aim was to distinguish between semantically related pairs from RuThes Lite thesaurus [Лукашевич 2011] and random pairings. Average precision was used as evaluation metrics for this track and for the tracks in the second task.

In the association task, participants had to detect whether the words or multiword expressions are associated (topically related) to each other. First track in this task mixed random pairings and associations taken from the Russian Associative Thesaurus[2]. The second track test set included associations from Sociation.org database[3].

An ideal system should have always assigned 0 to unrelated pairs and positive values to related or associated ones, thus achieving average precision of 1.0. In the case of the first semantic relatedness track an ideal system was to rank the pairs identically to the human judgment, to achieve Spearman's rho of 1.0.

In the end, participants were rated with four scores: **hj** (Spearman's rho for the first relatedness track), **rt** (average precision for the second relatedness track), **ae** (average precision for the first association track) and **ae2** (average precision for the second association track). The contest itself is described in detail in [Panchenko et al. 2015]. We participated in all tracks, using different models.

In general, the choice of test data and evaluation metrics seems to be sound. However, we would like to comment on two issues.

1. Test sets for the **rt** and **ae2** tasks include many related word pairs which share long character strings (e.g., "*благоразумие; благоразумность*"). This allows reaching unexpectedly good performance without building any complicated models, using only character-level analysis. We were able to achieve average precision of 0.79 for **rt** task and 0.72 for **ae2** task with the following algorithm: if two words share strings more than 3 characters in length, choose the longest of such strings; its length divided by 10 is the semantic similarity between words; if no such strings are found, assume similarity is zero.
   It seems trivial that in Russian, words which share stems are virtually always semantically similar in this or that way. Thus, the contest would benefit if the ratio of such pairs became lower, so that the participants had to design systems that strive to understand meaning, not to compare strings of characters. Certainly, this issue is conditioned by the usage of RuThes and Sociation databases, which by design contain lots of related words with common stems. It is difficult to design a dataset of semantically related lexical units for Russian which would not be haunted by this problem. However, this is the challenge for organizers of the future evaluations. Other RUSSE tracks do not suffer from this flaw.

---

[2] http://tesaurus.ru/dict/dict.php

[3] http://sociation.org

2. The test set for the **ae** task was Russian Associative Thesaurus. It was collected between 1988 and 1997; many entries can already be considered a bit archaic (*"колхоз; путь ильича"*, *"президент; ельцин"*, etc). Perhaps, this is the reason for often observed disagreement in systems' performance measured in **ae** and in **ae2**. These datasets differ chronologically, and it greatly influences association sets. Note striking difference in comparison to semantic relatedness task: synonymic, hyponymic and hypernymic relations are stable for dozens or even hundreds of years, while associations can dramatically change in ten years, depending on social processes. At the same time, such glitches cover only small part of the entries, and this is only a minor remark.

In the next chapter we describe our approach to computing semantic similarity for Russian.

3. Neural Networks Meet Corpora

The methods of automatically measuring semantic similarity fall into two large groups: knowledge-based and distributional ones [Harispe et al. 2013]. The former depend on building (manually or semi-automatically) a comprehensive ontology for a given language, which functions as a conceptual network. Once such a network is complete, one can employ various measures to calculate distance between concepts in this network: in general, the shorter is the path, the higher is the similarity.

We employed other, distributional approach, motivated by the notion that meaning is defined by usage and semantics can be derived from the contexts a given word takes [Lenci 2008]. Thus, these algorithms are inherently statistical and data-driven, not ruled by a curated conceptual system, as is the case for knowledge-based ones.

If lexical meaning is generally the sum of word usages, then the most obvious way to capture it is to take into account all contexts a word participates in, given a large enough corpus. In distributional semantics, words are usually represented as vectors in semantic space [Turney and Pantel 2010]. In other words, each lexical unit is a vector of its "neighborhood" to all other words in the lexicon, after applying various distances and weighting coefficients. The matrix of n rows and n columns (where n is the size of the lexicon) with "neighborhood degrees" in the cells is then a distributional model of the language. One can compare vectors for different words (e.g., calculating their cosine similarity) and find how "far" they are from each other. This distance turns out to be the semantic similarity we sought, expressed continuously from **0** (totally unrelated words) to **1** (absolute synonyms).

Such an approach theoretically scales well (one has to simply add more texts to the corpus to get new words and contexts) and does not demand laborious and subjective process of building an ontology. Meaning is extracted directly from linguistic evidence: the researcher only has to polish weighting algorithms. Also, fixed-length vector representations instead of orthographic words constitute excellent input to machine learning systems, independent of their particular aim.

The fly in the ointment is that traditional distributional semantic models (DSMs) are very computationally expensive. The reason is the dimensionality of their vectors,

generally equal to the size of the lexicon. As a result, a model has to operate on sparse but very large matrices. For example, if a corpus includes one million distinct word types (not a maximum value, as we show below), we will have to compute dot products of 1M-dimensional vectors each time we need to find how similar two words are. Vectors' dimensionality can be reduced to reasonable values using tricks like singular value decomposition or principal components, but this often degrades performance or quality.

As a kind of remedy to this, artificial neural networks can learn distributed vector representations or "neural embeddings" of comparatively small size (usually hundreds of components) [Bengio 2003]. Neural models are directly trained on large corpora to produce vectors which maximize similarity between contextual neighbors found in the data, while minimizing similarity for unseen contexts. Vectors are initialized randomly, but in the course of the training the model converges and semantically similar words obtain similar vector representations. However, these models were slow to train because of non-linear hidden layer.

Recently, **Continuous Bag-of-Words** (CBOW) and **Continuous Skip-gram** neural network language models without hidden layer, implemented in the *Word2Vec* tool [Mikolov et al. 2013a], seriously changed the field; using smart combination of already known techniques, they learn high quality embeddings in a very short time. These algorithms clearly outperform traditional DSMs in various semantic tasks [Baroni et al. 2014].

For this competition, we tested both CBOW and skip-gram models. Evaluation results (for a wide range of settings) are given in Section 4.

In order to train neural language models one needs not only algorithms, but also corpora. We used 3 text collections:

1. **News**: a corpus of contemporary Russian news-wire texts collected by a commercial news aggregator. Corpus volume is about 1.8 billion tokens, more than 19 million word types. It was crawled from 1500 news portals, and news pieces themselves are dated from 1 September of 2013 to 30 June of 2014 (more than 9 million documents total).
2. **Web**: a corpus of texts found on Russian web pages. It originates from a search index for one of the major search engines in the Russian market, thus is supposed to be quite representative. This source repository itself contains billions of documents, but to train the model we randomly selected about 9 million pieces (no attention was paid to their source or any other properties). Thus, hopefully the corpus contains all major types of texts found in the Internet, in nearly all possible genres and styles.
   Boilerplate and templates were filtered out to leave only main textual content of these pages, with the help of *boilerpipe* library [Kohlschütter et al. 2010]. After removing non-Cyrillic sentences, the resulting web corpus contained approximately 940 million tokens.
3. **Ruscorpora**: Russian National Corpus consists of texts which supposedly represent the Russian language as a whole. It has been developed for more than 10 years by a large group of top-ranking linguists, who select texts and segments for inclusion into the corpus. It was extensively described in the literature (see [Плунгян 2005], [Савчук 2005]). The size of the main part of RNC is 230 million word tokens, but we worked with the dump containing 174 million tokens.

All the corpora were lemmatized with *MyStem* [Segalovich 2003]. We used version 3.0 of the software, with disambiguation turned on. Stop-words were removed, as well as single-word sentences (they are useless for constructing context vectors). Because we removed stop-words ourselves, *word2vec* sub-sampling feature was not used. After this pre-processing, **News** corpus contained 1,300 million tokens, **Web** corpus 620 million tokens, and **Ruscorpora** 107 million tokens.

These corpora represent three different "stimuli" to neural network training algorithm. **Ruscorpora** is a balanced academic corpus of decent but comparatively small size, **Web** is large, but noisy and unbalanced. Finally, **News** is even larger than **Web**, but cleaner and biased towards one particular genre. These differences caused different results in semantic similarity tasks for models trained on the corpora in question (although all corpora proved to be good training sets).

We note that **Ruscorpora**, notwithstanding its size, certainly won this race, receiving scores essentially higher than the models trained on other two collections. The details are given in the next section.

4. Evaluation

There can be two reasons for a model to perform worse in comparison to the gold standard in this evaluation contest: either the model outputs incorrect similarity values (cosine distances in our case), or one or both words in the presented pair are unknown to the model. The former can be treated only by re-training the model with different settings or different training set, while the latter can be partially remedied by a couple of tricks, both of which we used.

The first trick exploits the issue described in the Section 2: many semantically similar words in Russian have common stems. We "computed" similarity using the longest common string algorithm in case of unknown words, as a kind of "emergency treatment". For **Ruscorpora** models it consistently increased average precision in **rt** track by 0.02 ... 0.05.

Another trick is building model assemblies, allowing to "fall back" to another model in case when unknown words are met. In our case, we knew that **Ruscorpora** model is the best, but only for the words it knows. The **Web** model is slightly worse, but knows a lot more distinct words (millions instead of hundreds of thousands). Thus, we query **Web** model for the word pairs unknown to **Ruscorpora**. Similarity measures range strictly from 0 to 1 and are generally compatible across models. Only if the words are unknown even to the **Web** model, we fall back further to the longest common string trick. In our experience, such assemblies seriously improved overall performance.

Most important training parameters for our task are algorithm, vector size, window size and frequency threshold. The algorithm can be either CBOW or skip-gram, with the latter being considerably slower. Also, skip-gram performance was consistently worse for all corpora except **news**. This seems to be specific for Russian, as previous research for English corpora stated that skip-gram is generally better [Mikolov 2013a].

Vector size is the number of dimensions in vector representations; increasing vector size generally increases both performance and training time. Window is context

width: how many words to the right and to the left will be considered. Larger window size increases training time and also leads to the model being more "topical" opposed to "functional" [Levy and Goldberg 2014]. It means that the model assigns similar vectors to topically associated words, not only to direct semantic relatives (synonyms, etc). This is quite natural, as the model trains on neighbors more distant from the analyzed lexical units. Unsurprisingly, models trained on large windows perform better in association tasks, while those trained on micro-windows of size 1 or 2 (only immediate neighbors) excel at catching direct semantic or functional relations.

Finally, frequency threshold or minimal count is a minimum frequency a word must possess in order to be considered by the model. All the lexical units with lower frequency are ignored during training and are not assigned vector representations. It is useful in order to get rid of low-frequency noise and train only on sufficiently presented evidence. Moreover, the less distinct words the model possess, the faster is training; the downside is, of course, absence of some words in the model lexicon.

In our experience, typical training speed on an Intel Xeon E5620 2.4GHz machine (14 cores) was 116,386 words per second for CBOW algorithm. Web corpus model training with vector size 500, minimal count 100, window 10 and 5 iterations (epochs) took approximately 7 hours; the model saw 3 168 819 885 words in total. This timing is consistent with [Mikolov et al. 2013a].

The Table 1 presents our best-performing models, as submitted to RUSSE contest.

Table 1. Our best results submitted to the evaluation

Track	**hj**	**rt**	**ae**	**ae2**
Rank (among 18participants)	2	5	5	4
Training settings	CBOW on **Ruscorpora** with context window 5, minimal count 5 + CBOW on **Web** with context window 10, minimal count 2	CBOW on **Ruscorpora** with context window 5, minimal count 5 + CBOW on **Web** with context window 10, minimal count 2	Skip-gram on **News** with context window 10, minimal count 10	CBOW on **Web** with context window 5, minimal count 2
Score	0.7187	0.8839	0.8995	0.9662

Note that minimal count values (defining how much of low-frequency long tail is cut off) are different for different corpora. The optimal setting possibly depends on the vocabulary distribution in a particular text collection, and on how closely it follows Zipfian law. We leave this for further research.

It is clear that **Ruscorpora** beats both **Web** and **News** corpus in the task of distinguishing semantically related words. This is impressive considering its size: it seems that balance and clever selection of texts for corpus do really make sense and allow the model to learn very high quality vectors. However, when we turn to the task

of detecting associations, sheer volume and diversity of **News** and **Web** become paramount, and they outperform **Ruscorpora** models. It is interesting that **News** model is better with predicting associations from Russian Associative Thesaurus. Probably, this reflects more "official" spirit of this resource in comparison with more colloquial nature of Sociaton.org database in the **ae2** track, better modeled with **Web** texts.

The plots below show how performance in different RUSSE tracks depends on training settings. Two parameters did not change: training mode (CBOW for **Ruscorpora** and **Web** and skip-gram for **News**) and minimal count (5 for **Ruscorpora**, 2 for **Web** and 10 for **News**); they reproduce the values in the Table 1. Only selected plots are shown here; see the link to the others in the Section 5.

The plots prove that while increasing vector size generally leads to quality increase, after a certain threshold this growth can sometimes stop or even revert[4]. This is the case for **Ruscorpora** (*Fig. 1*), but not for **Web** (*Fig. 2*) or **News**. We hypothesize that the reason is the size of these two corpora: the volume of data allows filling vector components with meaningful relationships, while with **Ruscorpora** the model can't learn so many relationships because of data insufficiency; as a result, vectors are filled with noise. This is again consistent with the notion that vector size increase must be accompanied by data growth, expressed in [Mikolov et al. 2013].

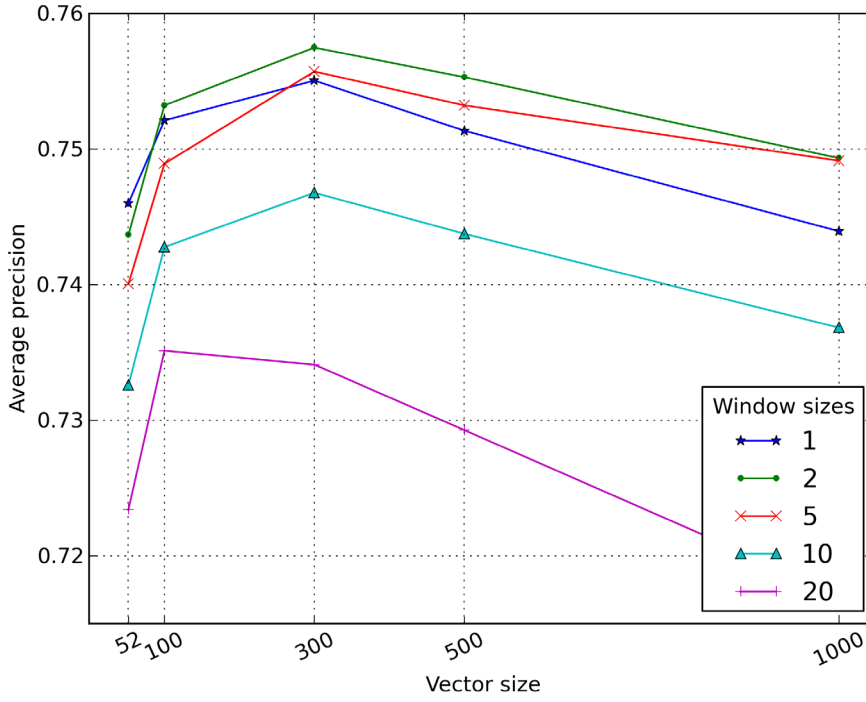


Fig. 1. Ruscorpora model performance in **rt** track depending on vector size

[4] Vector sizes start with 52, because training time is optimal when dimensionality is a multiple of 4.

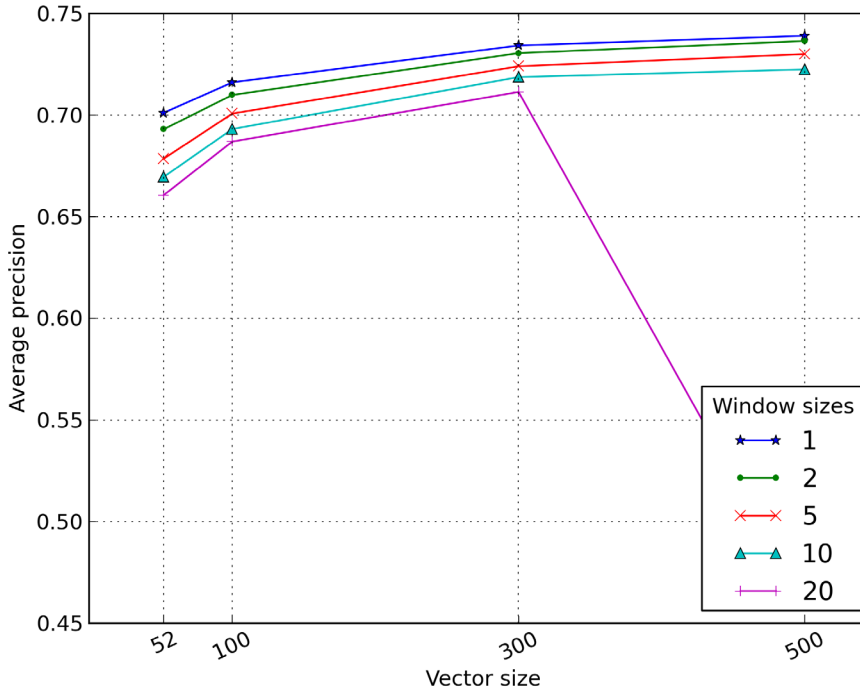


Fig. 2. Web model performance in **rt** track depending on vector size

As for the window size dynamics, we observe clear direct correlation between window size and **ae2** performance and inverse correlation for **rt** performance (*Fig. 3*). As already stated, a shorter window favors strict functional and semantic relations, while a larger window (10 words and more) allows catching more vague topical relations. Interestingly, **Ruscorpora** models are better at **ae** task with short windows, unlike **ae2** (*Fig. 4*); perhaps, associations from **ae** dictionary are more syntagmatic and tend to occur close to each other, while Sociation pairs are topical *par excellence*. This further proves deep difference between these two associative tasks.

5. Discussion

The first result of our research is that neural embedding models are shown to be directly applicable to Russian semantic similarity tasks. Rich morphology does not pose an obstacle for learning meaningful vector representations, with preprocessing limited to lemmatizing (training on unlemmatized text decreases performance, unlike English tasks where one often doesn't need to even stem the corpus). The result is very persuasive. We believe it is worth to try augmenting many NLP tools for Russian with neural embeddings to make existing instruments more semantically aware.

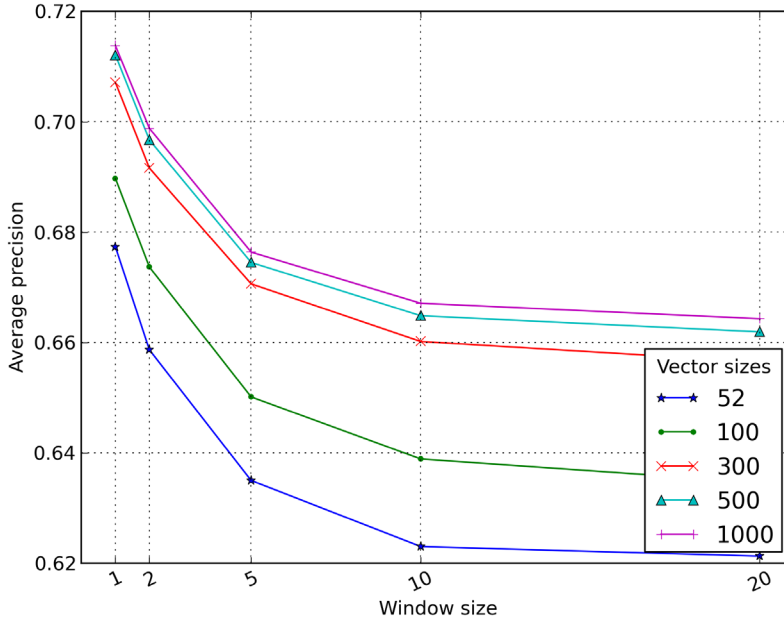


Fig. 3. News model performance in **rt** track depending on window size

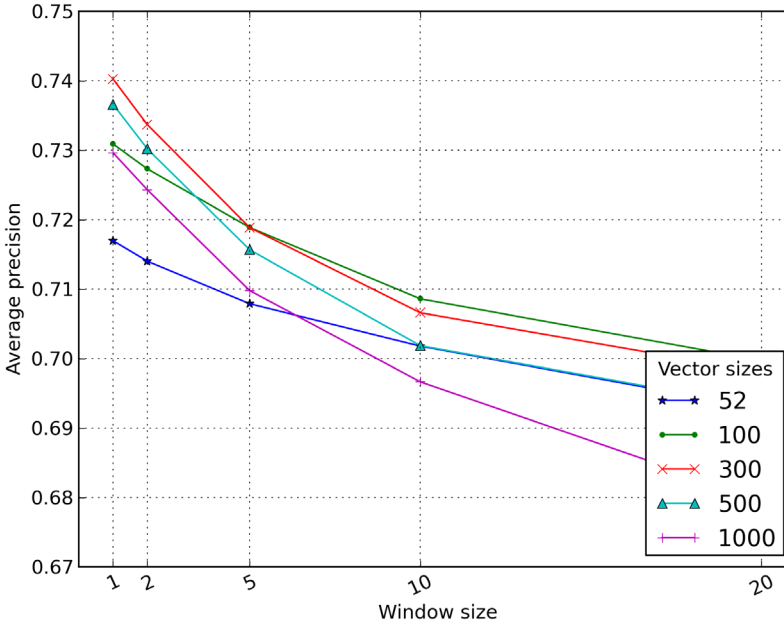


Fig. 4. Ruscorpora model performance in **ae** track depending on window size

Another, more unexpected outcome of our participation in RUSSE was that Russian National Corpus (RNC) turned out to be an excellent training set for neural network language models. When at start, we were sure that the amount of data plays dominant role and that the national corpus will eventually lose, because of being substantially smaller. However, it was quite the opposite: in the majority of comparisons (especially for semantic relatedness task) models trained on RNC outperformed their competitors, often even with vectors of lower dimensionality.

The only explanation is that RNC is really representative of the Russian language, thus providing balanced linguistic evidence for all major vocabulary tiers. Additionally, it seems to contain little or no noise and junk fragments, which sometimes occur in other corpora. To sum it up, we certainly recommend training neural language models on RNC, if this resource is available.

The resulting models for each of the three corpora, trained with optimal settings, can be downloaded at http://ling.go.mail.ru/misc/dialogue_2015.html; the full set of performance plots for different training settings is also there.

6. Future Work

We have only scratched the surface of exploiting neural embeddings to deal with Russian language material. The next step should be to perform a comprehensive study of errors typical for each model in their semantic similarity or other decisions. This can shed light on the real nature of differences between models and help in studying human errors.

Another very interesting field of research is corpora comparison through the output of neural language models trained on them [Kutuzov and Kuzmenko 2015]. Here we, in a way, arrive to an almost omnipotent "mind" able to rapidly evaluate huge corpora, taking into consideration what meanings words in their vocabularies take and how they are different from each other.

Of course, this is not an exhaustive outlook of computational linguistics research directions related to neural lexical vectors. Their foundational nature allows to employ them everywhere meaning is important; we anticipate a serious growth in semantic tools' quality.

Last but not least, we plan to implement a full-fledged web service for testing and querying distributed semantic models for Russian, particularly neural ones. A prototype to try with is already available online at http://ling.go.mail.ru/dsm.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments. Support from the Basic Research Program of the National Research University Higher School of Economics is also acknowledged.

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. *Bengio Y., Ducharme R., Vincent P., Janvin C.* (2003), A neural probabilistic language model, The Journal of Machine Learning Research, 3, pp. 1137–1155.
3. *Dinu G., Baroni M.* (2014), How to make words with vectors: Phrase generation in distributional semantics, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 624–633.
4. *Harispe S., Ranwez S., Janaqi S., Montmain J.* (2013), Semantic Measures for the Comparison of Units of Language, Concepts or Instances from Text and Knowledge Base Analysis, arXiv preprint, available at http://arxiv.org/abs/1310.1285
5. *Kohlschütter C., Fankhauser P., Nejdl W.* (2010), Boilerplate detection using shallow text features, Proceedings of the third ACM international conference on Web search and data mining, pp. 441–450.
6. *Kutuzov A., Kuzmenko E.* (2015), Comparing Neural Lexical Models of a Classic National Corpus and a Web Corpus: The Case for Russian, A. Gelbukh (Ed.): CICLing 2015, Part I, Springer LNCS 9041, pp. 47–58.
7. *Lenci A.* (2008), Distributional semantics in linguistic and cognitive research, Italian journal of linguistics, 20(1), pp. 1–31.
8. *Levy O., Goldberg Y.* (2014), Dependency-based word embeddings, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 2, pp. 302–308.
9. *Loukachevitch N. V.* (2011), Thesauri in information retrieval tasks [Тезаурусы в задачах информационного поиска], Moscow State University Press.
10. *Mikolov T., Chen K., Corrado G., Dean J.* (2013a). Efficient estimation of word representations in vector space, arXiv preprint, available at http://arxiv.org/abs/1301.3781
11. *Mikolov T., Le Q. V., Sutskever I.* (2013b). Exploiting similarities among languages for machine translation, arXiv preprint, available at http://arxiv.org/abs/1309.4168
12. *Panchenko A., Loukachevitch N. V., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015), RUSSE: The First Workshop on Russian Semantic Similarity, Proceedings of the Dialogue 2015 conference, Moscow, Russia, pp. xx-yy.
13. *Plungian V. A.* (2005), Why we make Russian National Corpus? [Зачем мы делаем Национальный корпус русского языка?], Otechestvennye Zapiski, 2
14. *Savchuk S. O.* (2005), Meta-text annotation in Russian National Corpus: foundations and main functions [Метатекстовая разметка в Национальном корпусе русского языка: базовые принципы и основные функции], Russian National Corpus, pp. 62–88.
15. *Segalovich I.* (2003), A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine, MLMTA, pp. 273–280.
16. *Turney P. D., Pantel, P.* (2010), From frequency to meaning: Vector space models of semantics, Journal of artificial intelligence research, 37(1), pp. 141–188.

ВЕКТОРНЫЕ МОДЕЛИ И ВСПОМОГАТЕЛЬНЫЕ МЕТОДЫ ДЛЯ ОПРЕДЕЛЕНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ СЛОВ РУССКОГО ЯЗЫКА

Лопухин К. А. (kostia.lopuhin@gmail.com)
ЧТД, Москва, Россия

Лопухина А. А. (nastya-merk@yandex.ru)
Институт русского языка им. В. В. Виноградова РАН, Москва, Россия

Носырев Г. В. (grigorij-nosyrev@yandex.ru)
Яндекс, Москва, Россия

Ключевые слова: семантическая близость, ассоциации, машинное обучение, семантические векторы, векторные модели

THE IMPACT OF DIFFERENT VECTOR SPACE MODELS AND SUPPLEMENTARY TECHNIQUES ON RUSSIAN SEMANTIC SIMILARITY TASK[1]

Lopukhin K. A. (kostia.lopuhin@gmail.com)
Chtd, Moscow, Russia

Lopukhina A. A. (nastya-merk@yandex.ru)
V. V. Vinogradov Russian Language Institute, Russian Academy of Sciences, Moscow, Russia

Nosyrev G. V. (grigorij-nosyrev@yandex.ru)
Yandex, Moscow, Russia

This paper presents a system for determining semantic similarity between words that was an entry for the Dialog 2015 Russian semantic similarity competition. The system introduced is primary based on word vector models, supplemented with various other methods, both corpus- and dictionary-based. In this paper we compare performance of two methods for building word vectors (word2vec and GloVe), evaluate how performance

[1] Работа выполнена при частичной финансовой поддержке Программы фундаментальных исследований Президиума РАН «Историческая память и российская идентичность» и гранта РГНФ №13-04-00307а.

varies on different corpus sizes and preprocessing techniques, and measure accuracy gains from supplementary methods. We compare system performance on word relatedness and word association tasks, and it turns out that different methods have varying relative importance for these tasks.

Key words: semantic similarity, associations, machine learning, semantic vectors, vector space model

1. Introduction

Semantic similarity is a measure of closeness of word meanings that can be represented as a number on some scale. The notion of semantic similarity includes different types of semantic relations: synonyms, hyponyms and hypernyms ("*свист*" (*whistle*), "*хрип*" (*wheeze*), "*стрекотня*" (*chirr*) and "*звук*" (*sound*); "*жвачка*" (*chewing gum*) and "*продукт*" (*product*); "*муж*" (*husband*) and "*мужчина*" (*man*)) and semantic associations, that link words by connotations ("*актер*" (*actor*) and "*игра*" (*performance*), "*грим*" (*make-up*); "*Айвазовский*" (*Ayvazovsky*) and "*маринист*" (*painter of seascapes*)). The last term, association, is loosely defined, and can range from pairs that average speaker might consider synonyms, to rather distant concepts.

Semantic similarity is an important building block in more complex natural language processing tasks, such as sentence and text similarity, machine translation [Mikolov et al 2013a], query expansion [Voorhees 1994], etc.

There are several approaches for determining semantic similarity: based on dictionaries, ontologies or machine learning. Synonym dictionaries are compiled manually and reflect human understanding of synonymy, but contain only one type of semantic relations and are deemed to be incomplete. Ontologies include hyponym relations and allow searching for the shortest connection between words or concepts, but also suffer from low recall. Machine learning solves low recall problem by training models on big corpora, but human understanding of semantic similarity is hard to model correctly.

2. Russian Semantic Similarity Evaluation (RUSSE)

Most approaches to semantic similarity were implemented and evaluated primarily in English, and there were no systematic evaluations of semantic similarity models for Russian until the RUSSE competition and workshop, held for Dialogue 2015 conference [Panchenko et al 2015][2]. Semantic similarity was measured on the following tracks:

- Human judgements track (hj): word similarity assessed by Russian native speakers.
- Relatedness track (rt): relations sampled from RuThesLite Tesaurus.
- First association track (ae): relations sampled from Russian Associative Thesaurus.
- Second association track (ae2): relations sampled from Sociation.org online experiment.

[2] http://russe.nlpub.ru

Evaluation metric for human judgements track was Spearman's rank correlation, and AUC under the ROC curve for the other tracks.

In this paper we describe a system that was an entry for RUSSE competition and analyse its performance.

3. Word vector models

One of the most widely used machine learning approaches for determining semantic similarity is building word vector models from large corpora and using distance in this vector space as a measure of semantic similarity. Word vector models represent each word as a low-dimensional (50–1,000 components) vector, built based on words contexts in corpus.

These models are often called semantic vector space models, because components of the vectors exhibit semantic properties [Mikolov et al 2013b]: for example, the difference between vectors for "*king*" and "*queen*" is very close to the difference of "*man*" and "*woman*". The most useful property for our task is that semantically similar words have similar vectors. Word similarity is usually defined as a cosine of the angle between two word vectors (cosine similarity).

We decided to use word vectors for modelling word similarity because they are known to perform well for this task [Mikolov et al 2013c] and are straightforward to implement. Another benefit is that they give continuous similarity measure out of the box, which is useful for hj track and simplifies augmentation with other models.

There are several different algorithms for computing word vectors. In this paper we evaluated word2vec skip-gramm algorithm [Mikolov et al 2013c] using gensim implementation and GloVe [Pennington et al 2014] algorithm using reference implementation. Some studies [Shi at al 2014] suggest that although these two algorithms have quite different numerical formulation, their optimization objectives are similar. But in practice these algorithms produce vectors which quality very much depends on the task at hand. In our case it turns out that word2vec models perform better on all tracks, as we can see in the following table:

Table 1. Comparison of word2vec and GloVe models

	word2vec	**GloVe**	**ratio**
hj	**0.76254**	0.66537	14.60%
rt	**0.92277**	0.90128	2.38%
ae	**0.95525**	0.95427	0.10%
ae2	**0.98354**	0.97723	0.65%

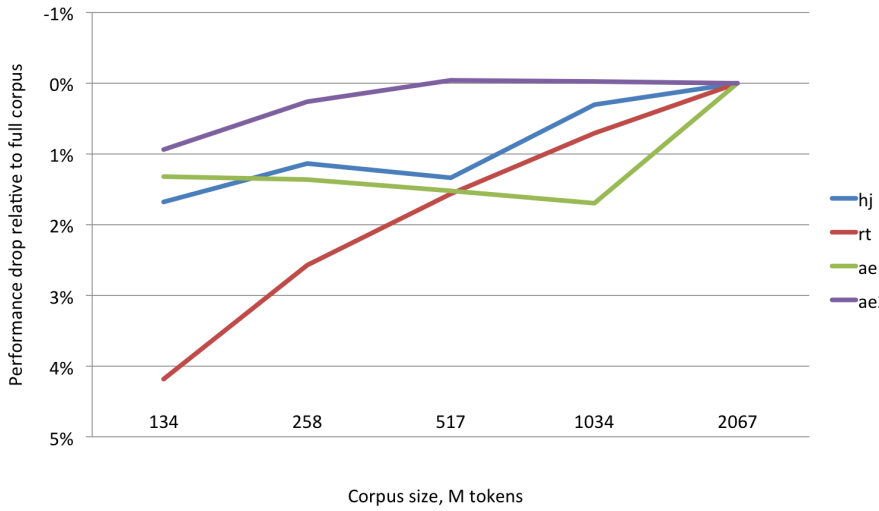
Note that we did not do extensive meta-parameter optimization: we used window size 10, and vector size 300, leaving other parameters at default values. We used cosine similarity for both methods, although there might be a better measure, especially in the case of GloVe.

4. Importance of corpora size and preprocessing

Quality of corpus-based models usually depends on the size and quality of the corpus and preprocessing techniques. Knowing that, we used the biggest corpus we could get at the time, by combining several separate corpora: ruwac[3] (1,268 M tokens), lib.ru (624 M tokens), and Russian Wikipedia[4] (176 M tokens). Even for such a large corpus rare words were still a problem, so we used a rather low frequency cutoff of 10, which gave us vocabulary size 844,530. In order to measure how model quality depends on corpus size, we compared final system performance on randomly sampled sub-corpora of various sizes. Results are represented as a table, that shows performance loss relative to full corpus.

Table 2. Impact of corpus size

rel. size	0.5	0.25	0.125	0.065
hj	0.31%	1.34%	1.13%	1.68%
rt	0.71%	1.57%	2.58%	4.19%
ae	1.70%	1.52%	1.36%	1.32%
ae2	−0.02%	−0.04%	0.27%	0.94%



This suggests that increasing corpus size might be worthwhile for most tracks.

Model and corpus building time should also be considered. We needed 4 hours for corpus preprocessing and 8 hours for model training using 8 cores for the full corpus.

[3] http://corpus.leeds.ac.uk/tools/ru/ruwac-parsed.out.xz

[4] https://s3-eu-west-1.amazonaws.com/dsl-research/wiki/wiki-ru-noxml.txt.bz2

Besides basic preprocessing (getting rid of html markup, short sentences, etc.) we also experimented with using lemmatizer as a preprocessing step. On one hand, we lose valuable grammatical information here, so the quality of the vectors might decrease. On the other hand, lemmatizing helps mitigate low frequency words problem and allows comparing lemmas and not word forms.

As we see in the following table, lemmatizing hugely influences human judgements track performance and is also important for other tracks.

Table 3. Impact of lemmatization

	lem	no lem	ratio
hj	**0.76254**	0.60014	27.06%
rt	**0.92277**	0.86150	7.11%
ae	**0.95525**	0.91079	4.88%
ae2	**0.98354**	0.94570	4.00%

So far we have described the base of our method: word vector model built with word2vec on a large corpus with lemmatization.

5. Supplementary models and sources

The first association track (ae) contained a certain number of high frequency bigrams, like "*человек*" (*man*) and "*амфибия*" (*amphibian*) or "*время*" (*time*) and "*не ждет*" (*does not wait*), so **bigram model** was used to supplement the word vector model. Bigram model was built from the same corpus that was used for word vectors, but with stop words (prepositions, conjunction, etc.) removed. In order to convert bigram score into [0, 1] range, we used ad hoc normalized PMI: $log\,(max(1{,}1 + PMI))\,/\,2$. Bigram model was used only on ae and ae2 tracks, with ae gaining 7.49%, and ae2 just 0.89%. On hj and rt tracks performance with bigram model dropped significantly, up to 7.84% for hj track.

Analysis of errors on training datasets revealed two major sources of errors:

1. Low frequency words: some words, especially in rt training dataset, were never seen in the corpus, for example "*автохтонка*" (*woman-indigene*), "*магометанство*" (*Mohammedanism*).
2. High frequency words having common semantic components, but not synonyms or hyponyms: such words are often used in similar contexts, and thus have high similarity according to word vector model, for example "*собрат*" (*brother*) and "*предшественник*" (*predecessor*) or "*блузочка*" (*blouse*) and "*платьице*" (*dress*).

Such errors are hard to resolve with just word vector and bigram models, so we introduced a number of supplementary models and sources to overcome them:

- synonyms database
- prefix database
- orthographic similarity model
- secondary orthographic similarity model
- hyphen handling

They are described in more detail below.

Synonyms database is a database of synonyms compiled from five dictionaries[5] by students and researchers from the Higher School of Economics. Synonyms are given for 43,679 words, constituting 135,134 pairs, as many words have several synonyms. If word A had synonyms S1..Sn, then pairs (Si, Sk) were also considered synonyms, but with a lower weight—such extension gave 1,556,374 word pairs. Recall on rt train dataset is 7.64%, with 0.63% false positives. False positive ratio is the number of cases where model considered words as similar, divided by the total number of predictions by the model (and not by the total number of pairs in training set). Gain from this model (how much precision dropped when dropping this model from the final method) ranged from 1.53% to −0.03% on different tracks within the test dataset, with maximum gain on human judgements track. Synonym databases should be used if possible, as they are very easy to incorporate into existing models and increase performance without significant drawbacks.

Prefix database is a list of greek and latin prefixes, extracted from "The anatomy of terms. 400 derivation elements from Latin and Greek" [Bykov 2008], that give strong contribution to the word meaning, like "auto", "aero", etc. If two words shared such prefix, they were considered similar. This model was added to overcome low frequency words problem for pairs such as "*авиаконцерн*" (*aviaconcern*) and "*авиаконсорциум*" (*aviaconsortium*). Recall on rt train dataset is 0.82%, with 0.53% false positives. The only track that gained a little from this model was rt track, with 0.15% gain. Despite such a low gain, we still used it in the competition, but generally this model seems to be of little use due to very low recall.

Orthographic similarity model measures similarity in spelling, and improves handling of low frequency word pairs like "*автохтон*" (*indigene*) and "*автохтонка*" (*woman-indigene*). More precisely, it searches for a longest common beginning or ending, and then gives similarity in [0, 1] range based on its length and lengths of compared words. It is especially useful in case of two cognate words of different gender ("*агроном*" (*agriculturist*) and "*агрономша*" (*woman-agriculturist*)), or usage of some rare stem ("*авангардность*" (*vanguardness*) and "*авангардизм*" (*avant-gardism*)). Such cases could also be handled by stemming.

Recall for this model on rt train dataset is 6.40%, with 1.76% false positives. Gain from this model, combined with secondary similarity model is up to 1.55% for rt track. Due to our definition of gain, we can not measure the gain without secondary similarity model, but we can compare the gain against pure word2vec model: it is 0.58% for rt track.

Secondary orthographic similarity model extends the gains in orthographic similarity model to more cases. For example, words "*водитель*" (*driver*) and "*автолюбительница*" (*woman-motorist*) are not considered similar by the model, because "*автолюбительница*" (*woman-motorist*) is absent from the word vector model. But we have a pair "*водитель*" (*driver*) and "*автолюбитель*" (*motorist*), where words are similar according to word vector model, and a pair "*автолюбитель*" (*motorist*) and

[5] http://web-corpora.net/synonyms

"*автолюбительница*" (*woman-motorist*), where words are similar according to orthographic similarity model. Secondary model can thus infer that the original pair "*водитель*" (*driver*) and "*автолюбительница*" (*woman-motorist*) has high similarity, namely the multiplication of two other similarity measures. Recall on rt train dataset is 7.20% (that is, ratio of pairs that gained higher similarity measure). Gain from this model is 1.00% for rt track.

Hyphen handling was added to improve similarity assessment of words like "*компания-монополист*» (*monopolist company*), "*писатель-фантаст*» (*science fiction writer*) that are rather rare by itself, but are composed of high-frequency words. This handling is very primitive: words are split by hyphen, and all possible pairs are compared for similarity, e.g. for pair "*предпринимательство*» (*enterprise*) and "*кибер-коммерция*» (*cyber-commerce*) the resulting pairs would be "*предпринимательство*» (*enterprise*), "*кибер*» (*cyber*) and "*предпринимательство*» (*enterprise*), "*коммерция*» (*commerce*). Obviously, words with hyphens constitute a small fraction of all words, so recall is only 1.11%, and gain from this special handling is only 0.10% for hj and rt tracks.

Models described in this section have low recall and very low false positive rate, and each returns normalized score in [0; 1] range, so we used the **maximum** of model predictions in the combined model. In order to quantify the gains from separate models, we measured system performance with each model removed, and also measured performance of word vector model without any additional models. Overall, we can summarize gains from the models in the following table (each cell contains performance relative to the full model). Note that bigram model is used for all figures in italic (ae and ae2 tracks except the second column).

Table 4. Performance drops when excluding supplementary models

	full with bigrams	**without bigrams**	**without synonyms**	**without prefix**	**without 2nd. orth. sim.**	**without orth. sim.**	**without hyphen**	**only word2vec**
hj	7.84%	**0.00%**	1.53%	0.00%	0.19%	0.26%	0.10%	1.78%
rt	0.47%	**0.00%**	0.64%	0.15%	1.00%	1.55%	0.10%	2.41%
ae	*0.00%*	7.49%	*0.13%*	*0.03%*	*0.00%*	*0.04%*	*0.01%*	***−0.16%***
ae2	*0.00%*	0.89%	***−0.05%***	*0.01%*	*0.00%*	*0.05%*	*0.01%*	*0.08%*

As we can see, apart from bigram model in case of ae track, other models give modest performance gains, especially on ae and ae2 tracks. Still, combining all models gives around 2% of improvement for hj and rt tracks.

In the case of determining synonymy and hyponymy, supplementary models and sources (namely, synonyms database and orthographic similarity) improve overall performance. In the case of associations we did not find any useful additional sources or techniques, and just a combination of word2vec and bigram models gives the best result.

6. Conclusion and future work

We presented a system for determining semantic similarity between Russian words. The system was developed in Python and is free to download and use[6].

We compared two vector models, analysed the importance of lemmatization and corpus size, and measured the gain of supplementary models. It turned out that word vector model gives the main contribution for word similarity task, and it can be successfully enhanced with other techniques tailored to the task at hand.

We think that further development is possible, and improvement of word vector model seems to be the most promising approach. Most obvious things to try would be increasing corpus size, tuning meta-parameters, experimenting with other solutions to different word forms problem (the one we solved with lemmatizing here). It could be also useful to understand the reason for relatively poor performance of GloVe model.

Another area we did not touch here is the nature of the task in which semantic similarity is needed, as it is not the end in itself. Such external context could influence system design. These types of models also seem a promising start for the problem of word sense disambiguation, as an extension of work on the word sense frequency database [Iomdin et al 2014]. The model might serve a basis for computing context vectors and, by clustering them, derive the senses of polysemantic words.

Acknowledgments

We would like to thank our friends and colleagues Anna Vybornova and Maria Kartysheva for valuable advice and resources. Synonyms database was kindly provided by Valentina Apresjan and HSE students.

References

1. *Bykov A. A.* (2008), The anatomy of terms. 400 derivation elements from Latin and Greek [Anatomiya terminov. 400 slovoobrazovatelnykh elementa iz latyni i grecheskogo], ENAS, Moscow.
2. *Iomdin B. L., Lopukhina A. A., Nosyrev G. V.* (2014), Towards a word sense frequency dictionary [K sozdaniyu chastotnogo slovarya znacheniy slov], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2014" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2014"], Bekasovo, pp. 199–212.
3. *Tomas Mikolov, Quoc V. Le, Ilya Sutskever.* (2013a) Exploiting Similarities among Languages for Machine Translation. arXiv:1309.4168 [cs.CL]

[6] https://bitbucket.org/kostialopuhin/russe

4. *Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Je Dean.* (2013) Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems (NIPS).
5. *Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.* (2013c) Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR.
6. *Panchenko A., Loukachevitch N. V., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015) RUSSE: The First Workshop on Russian Semantic Similarity. In Proceeding of the Dialogue 2015 conference. Moscow, Russia (in print).
7. *Jerey Pennington, Richard Socher, and Christopher D Manning.* (2014b) Glove: Global vectors for word representation. In Conference on Empirical Methods on Natural Language Processing (EMNLP).
8. *Tianze Shi, Zhiyuan Liu.* (2014) Linking GloVe with word2vec. arXiv:1411.5595 [cs.CL]
9. *Ellen M. Voorhees.* (1994), Query Expansion using Lexical-Semantic Relations. SIGIR '94.

Abstracts

SENTIRUEVAL: TESTING OBJECT-ORIENTED SENTIMENT ANALYSIS SYSTEMS IN RUSSIAN

Loukachevitch N. V. (louk_nat@mail.ru)[1], **Blinov P. D.** (blinoff.pavel@gmail.com)[2], **Kotelnikov E. V.** (kotelnikov.ev@gmail.com)[2], **Rubtsova Y. V.** (yu.rubtsova@gmail.com)[3], **Ivanov V. V.** (nomemm@gmail.com)[4], **Tutubalina E.** (tlenusik@gmail.com)[4]
[1]Lomonosov Moscow State University, Moscow, Russia;
[2]Vyatka State Humanities University, Kirov, Russia;
[3]A. P. Ershov Institute of Informatics Systems, Novosibirsk, Russia;
[4]Kazan Federal University, Kazan, Russia

The paper describes the data, rules and results of SentiRuEval, evaluation of Russian object-oriented sentiment analysis systems. Two tasks were proposed to participants. The first task was aspect-oriented analysis of reviews about restaurants and automobiles, that is the primary goal was to find word and expressions indicating important characteristics of an entity (aspect terms) and then classify them into polarity classes and aspect categories. The second task was the reputation-oriented analysis of tweets concerning banks and telecommunications companies. The goal of this analysis was to classify tweets in dependence of their influence on the reputation of the mentioned company. Such tweets could express the user's opinion or a positive or negative fact about the organization.

SYNTAX-BASED SENTIMENT ANALYSIS OF TWEETS IN RUSSIAN

Adaskina Yu. V. (adaskina@gmail.com), **Panicheva P. V.** (ppolin86@gmail.com), **Popov A. M.** (hedgeonline@gmail.com), InfoQubes, Moscow, Saint Petersburg, Russia

The paper describes our approach to the task of sentiment analysis of tweets within SentiRuEval—an open evaluation of sentiment analysis systems for the Russian language. We took part in the task of object-oriented sentiment analysis of Russian tweets concerning two types of organizations: banks and telecommunications companies. On both datasets, the participants were required to perform a three-way classification of tweets: positive, negative or neutral.

We used various statistical methods as basis for our machine learning algorithms and checked which features would provide the best results. Syntactic relations proved to be a crucial feature to any statistical method evaluated, but SVM-based classification performed better than the others. Normalized words are another important feature for the algorithm.

The evaluation revealed that our method proved to be rather successful: we scored the first in three out of four evaluation measures.

SEMANTIC SIMILARITY FOR ASPECT-BASED SENTIMENT ANALYSIS

Blinov P. D. (blinoff.pavel@gmail.com), **Kotelnikov E. V.** (kotelnikov.ev@gmail.com), Vyatka State Humanities University, Kirov, Russian Federation

The paper investigates the problem of automatic aspect-based sentiment analysis. Such version is harder to do than general sentiment analysis, but it significantly pushes forward the limits of unstructured text analysis methods. In the beginning previous approaches and works are reviewed. That part also gives data description for train and test collections.

In the second part of the article the methods for main subtasks of aspect-based sentiment analysis are described. The method for explicit aspect term extraction relies on the vector space of distributed representations of words. The term polarity detection method is based on use of pointwise mutual information and semantic similarity measure. Results from SentiRuEval workshop for automobiles and restaurants domains are given. Proposed methods achieved good results in several key subtasks. In aspect term polarity detection task and sentiment analysis of whole review on aspect categories methods showed the best result for both domains. In the aspect term categorization task our method was placed at the second position. And for explicit aspect term extraction the first result obtained for the restaurant domain according to partial match evaluation criteria.

EXTRACTING ASPECTS, SENTIMENT AND CATEGORIES OF ASPECTS IN USER REVIEWS ABOUT RESTAURANTS AND CARS

Ivanov V. V. (nomemm@gmail.com), **Tutubalina E. V.** (tutubalinaev@gmail.com),
Mingazov N. R. (nicrotek547@gmail.com), **Alimova I. S.** (alimovaIlseyar@gmail.com),
Kazan Federal University, Kazan, Russia

This paper describes a method for solving aspect-based sentiment analysis tasks in restaurant and car reviews subject domains. These tasks were articulated in the Sentiment Evaluation for Russian (SentiRuEval-2015) initiative. During the SentiRuEval-2015 we focused on three subtasks: extracting explicit aspect terms from user reviews (tasks A), aspect-based sentiment classification (task C) as well as automatic categorization of aspects (task D).

In aspect-based sentiment classification (tasks C and D) we propose two supervised methods based on a Maximum Entropy model and Support Vector Machines (SVM), respectively, that use a set of term frequency features in a context of the aspect term and lexicon-based features. We achieved 40% of macro-averaged F-measure for cars and 40,05% for reviews about restaurants in task C. We achieved 65.2% of macro-averaged F-measure for cars and 86.5% for reviews about restaurants in task D. This method ranked first among 4 teams in both subject domains. The SVM classifier is based on unigram features and pointwise mutual information to calculate category-specific score and associate each aspect with a proper category in a subject domain.

In task A we carefully evaluated performance of a method based on syntactic and statistical features incorporated in a Conditional Random Fields model. Unfortunately, the method did not show any significant improvement over a baseline. However, its results are also presented in the paper.

A HIGH PRECISION METHOD FOR ASPECT EXTRACTION IN RUSSIAN

Mayorov V. (vmayorov@ispras.ru), **Andrianov I.** (ivan.andrianov@ispras.ru),
Astrakhantsev N. (astrakhantsev@ispras.ru), **Avanesov V.** (avanesov@ispras.ru),
Kozlov I. (kozlov-ilya@ispras.ru), **Turdakov D.** (turdakov@ispras.ru),
Institute for System Programming of RAS, Moscow, Russia

This paper presents a work carried out by ISPRAS on aspect extraction task at SentiRuEval 2015. Our team submitted one run for Task A and Task B and got best precision for both tasks for all domains among all participants. Our method also showed the best F1-measure for exact aspect term matching for task A for automobile domain and both for Task A and Task B for restaurant domain.

The method is based on sequential classification of tokens with SVM. It uses local, global, syntactic-based, GloVe, topic modeling and automatic term recognition features. In this paper we also present evaluation of significance of different feature groups for the task.

AUTOMATIC OBJECT-ORIENTED SENTIMENT ANALYSIS BY MEANS OF SEMANTIC TEMPLATES AND SENTIMENT LEXICON DICTIONARIES

Polyakov P. Yu. (pavel@rco.ru), **Kalinina M. V.** (kalinina_m@rco.ru),
Pleshko V. V. (vp@rco.ru), RCO LLC, Moscow, Russia

This paper studies use of a linguistics-based approach to automatic object-oriented sentiment analyses. The original task was to extract users' opinions (positive, negative, neutral) about telecom companies, expressed in tweets and news. We excluded news from the dataset because we believe that formal texts significantly differ from informal ones in structure and vocabulary and therefore demand a different approach. We confined ourselves to the linguistic approach based on syntactic and semantic analysis. In this approach a sentiment-bearing word or expression is linked to its target object at either of two stages, which perform successively. The first stage includes usage of semantic templates matching the dependence tree, and the second stage involves heuristics for linking sentiment expressions and their target objects when syntactic relations between them do not exist. No machine learning was used. The method showed a very high quality, which roughly coincides with the best results of machine learning methods and hybrid approaches (which combine machine learning with elements of syntactic analysis).

DEEP RECURRENT NEURAL NETWORKS FOR MULTIPLE LANGUAGE ASPECT-BASED SENTIMENT ANALYSIS OF USER REVIEWS

Tarasov D. S. (dtarasov3@gmail.com), Reviewdot research,Kazan,Russian Federation

Deep Recurrent Neural Networks (RNNs) are powerful sequence models applicable to modeling natural language. In this work we study applicability of different RNN architectures including uni- and bi-directional Elman and Long Short-Term Memory (LSTM) models to aspect-based sentiment analysis that includes aspect terms extraction and aspect term sentiment polarity prediction tasks. We show that single RNN architecture without manual feature-engineering can be trained to do all these subtasks on English and Russian datasets. For aspect-term extraction subtask our system outperforms strong Conditional Random Fields (CRF) baselines and obtains state-of-the-art performance on Russian dataset. For aspect terms polarity prediction our results are below top-performing systems but still good for many practical applications.

A SUPERVISED APPROACH FOR SENTIRUEVAL TASK ON SENTIMENT ANALYSIS OF TWEETS ABOUT TELECOM AND FINANCIAL COMPANIES

Tutubalina E. V. (tutubalinaev@gmail.com)[1], **Zagulova M. A.** (mazagulova@stud.kpfu.ru)[1], **Ivanov V. V.** (nomemm@gmail.com)[1, 2], **Malykh V. A.** (valentin.malykh@phystech.edu)[3]
[1]Kazan Federal University (KFU), Kazan, Russia
[2]Institute of Informatics, Tatarstan Academy of Sciences, Kazan, Russia
[3]Institute for Systems Analysis RAS, Moscow, Russia

This paper describes a supervised approach for solving a task on sentiment analysis of tweets about banks and telecom operators. The task was articulated as a separate track in the Sentiment Evaluation for Russian (SentiRuEval-2015) initiative. The approach we proposed and evaluated is based on a Support Vector Machine model that classifies sentiment polarities of tweets. The set of features includes term frequency features, twitter-specific features and lexicon-based features. Given a domain, two types of sentiment lexicons were generated for feature extraction: (i) manually created lexicons, constructed from *Pros* and *Cons* reviews; (ii) automatically generated lexicons, based on pointwise mutual information between unigrams in a training set.

In the paper we provide results of our method and compare them to results of other teams participated in the track. We achieved 35.2% of macro-averaged F-measure for banks and 44.77% for tweets about telecom operators. The method described in the paper is ranked second and fourth among 7 and 9 teams, respectively. The best SVM setting after tuning parameters of the classifier and error analysis with common types of errors are also presented in this paper.

ASPECT EXTRACTION AND TWITTER SENTIMENT CLASSIFICATION BY FRAGMENT RULES

Vasilyev V. G. (vvg_2000@mail.ru), **Denisenko A. A.** (denisenko_alec@mail.ru), **Solovyev D. A.** (dmitry_soloviev@bk.ru), OOO «LAN-PROJECT», Moscow, Russia

The paper deals with approaches to explicit aspect extraction from user reviews of restaurants and sentiment classification of Twitter messages of telecommunication companies based on fragment rules. This paper presents fragment rule model to sentiment classification and explicit aspect extraction. Rules may be constructed manually by experts and automatically by using machine learning procedures. We propose machine learning algorithm for sentiment classification which uses terms that are made by fragment rules and some rule based techniques to explicit aspect extraction including a method based on filtration rule generation. The article presents the results of experiments on a test set for twitter sentiment classification of telecommunication companies and explicit aspect extraction from user review of restaurant. The paper compares the proposed algorithms with baseline and the best algorithm to track. Training sets, evaluation metrics and experiments are used according to SentiRuEval. As our future work, we can point out such directions as: applying semi-supervised methods for rule generation to reduce the labor cost, using active learning methods, constructing a visualization system for rule generation, which can provide the interaction process with experts.

RUSSE: THE FIRST WORKSHOP ON RUSSIAN SEMANTIC SIMILARITY

Panchenko A. (panchenko@lt.informatik.tu-darmstadt.de), TU Darmstadt, Darmstadt, Germany, Université catholique de Louvain, Louvain-la-Neuve, Belgium; **Loukachevitch N. V.** (louk_nat@mail.ru), Moscow State University, Moscow, Russia; **Ustalov D.** (dau@imm.uran.ru), N. N. Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the RAS, Russia; NLPub, Yekaterinburg, Russia; **Paperno D.** (denis.paperno@unitn.it), University of Trento, Rovereto, Italy; **Meyer C. M.** (meyer@ukp.informatik.tu-darmstadt.de), TU Darmstadt, Darmstadt, Germany; **Konstantinova N.** (n.konstantinova@wlv.ac.uk), University of Wolverhampton, Wolverhampton, UK

The paper gives an overview of the Russian Semantic Similarity Evaluation (RUSSE) shared task held in conjunction with the Dialogue 2015 conference. There exist a lot of comparative studies on semantic similarity, yet no analysis of such measures was ever performed for the Russian language. Exploring this problem for the Russian language is even more interesting, because this language has features, such as rich morphology and free word order, which make it significantly different from English, German, and other well-studied languages. We attempt to bridge this gap by proposing a shared task on the semantic similarity of Russian nouns. Our key contribution is an evaluation methodology based on four novel benchmark datasets for the Russian language. Our analysis of the 105 submissions from 19 teams reveals that successful approaches for English, such as distributional and skip-gram models, are directly applicable to Russian as well. On the one hand, the best results in the contest were obtained by sophisticated supervised models that combine evidence from different sources. On the other hand, completely unsupervised approaches, such as a skip-gram model estimated on a large-scale corpus, were able score among the top 5 systems.

EVALUATING THREE CORPUS-BASED SEMANTIC SIMILARITY SYSTEMS FOR RUSSIAN

Arefyev N. V. (narefjev@cs.msu.su), Lomonosov Moscow State University & Digital Society Laboratory, Moscow, Russia; **Panchenko A. I.** (panchenko@lt.informatik.tu-darmstadt.de), TU Darmstadt, Darmstadt, Germany; **Lukanin A. V.** (artyom.lukanin@gmail.com), LLC "SoftPlus", Chelyabinsk, Russia; **Lesota O. O.** (cheesemaid@gmail.com), Lomonosov Moscow State University, Moscow, Russia; **Romanov P. V.** (romanov4400@gmail.com) 1C Company, Moscow, Russia

This paper reports results of our participation in the first shared task on Russian Semantic Similarity Evaluation (RUSSE). We compare three corpus-based systems that measure semantic similarity between words. The first one uses lexico-syntactic patterns to retrieve sentences indicating a particular semantic relation between words. The second one builds traditional context window approach on the top of Google N-Grams data to take advantage of the huge corpora it was collected on. The third system uses *word2vec* trained on a huge *lib.rus.ec* book collection. *word2vec* is one of the state-of-the-art methods for English. Our initial experiments showed that it yields the best results for Russian as well, comparing to other two systems considered in this paper. Therefore, we focus on study of *word2vec* meta-parameters and investigate how the training corpus affects quality of produced word vectors. Finally, we propose a simple but useful technique for dealing with out-of-vocabulary words.

USING FOLKSONOMY DATA FOR DETERMINING SEMANTIC SIMILARITY

Klyachko E. (elenaklyachko@gmail.com), Moscow, Russia

This paper presents a method for measuring semantic similarity. Semantic similarity measures are important for various semantics-oriented natural language processing tasks, such as Textual Entailment or Word Sense Disambiguation. In the paper, a folksonomy graph is used to determine the relatedness of two words. The construction of a folksonomy from a collaborative photo tagging resource is described. The problems which occur during the process are analyzed and solutions are proposed. The structure of the folksonomy is also analyzed. It turns out to be a social network graph. Graph features, such as the path length, or the Jaccard similarity coefficient, are the input parameters for a machine learning classifying algorithm. The comparative importance of the parameters is evaluated. Finally, the method was evaluated in the RUSSE evaluation campaign. The results are lower than most results for distribution-based vector models. However, the model itself is cheaper to build. The failures of the models are analyzed and possible improvements are suggested.

TEXTS IN, MEANING OUT: NEURAL LANGUAGE MODELS IN SEMANTIC SIMILARITY TASKS FOR RUSSIAN

Kutuzov A. (akutuzov@hse.ru), National Research University Higher School of Economics and Mail.ru Group, Moscow, Russia; **Andreev I.** (i.andreev@corp.mail.ru), Mail.ru Group, Moscow, Russia

Distributed vector representations for natural language vocabulary get a lot of attention in contemporary computational linguistics. This paper summarizes the experience of applying neural network language models to the task of calculating semantic similarity for Russian. The experiments were performed in the course of Russian Semantic Similarity Evaluation track, where our models took from 2nd to 5th position, depending on the task.

We introduce the tools and corpora used, comment on the nature of the evaluation track and describe the achieved results. It was found out that Continuous Skip-gram and Continuous Bag-of-words models, previously successfully applied to English material, can be used for semantic modeling of Russian as well. Moreover, we show that texts in Russian National Corpus (RNC) provide an excellent training material for such models, outperforming other, much larger corpora. It is especially true for semantic relatedness tasks (although stacking models trained on larger corpora on top of RNC models improves performance even more).

High-quality semantic vectors learned in such a way can be used in a variety of linguistic tasks and promise an exciting field for further study.

THE IMPACT OF DIFFERENT VECTOR SPACE MODELS AND SUPPLEMENTARY TECHNIQUES ON RUSSIAN SEMANTIC SIMILARITY TASK

Lopukhin K. A. (kostia.lopuhin@gmail.com), Chtd, Moscow, Russia;
Lopukhina A. A. (nastya-merk@yandex.ru), V. V. Vinogradov Russian Language Institute, Russian Academy of Sciences, Moscow, Russia; **Nosyrev G. V.** (grigorij-nosyrev@yandex.ru), Yandex, Moscow, Russia

This paper presents a system for determining semantic similarity between words that was an entry for the Dialog 2015 Russian semantic similarity competition. The system introduced is primary based on word vector models, supplemented with various other methods, both corpus- and dictionary-based. In this paper we compare performance of two methods for building word vectors (word2vec and GloVe), evaluate how performance varies on different corpus sizes and preprocessing techniques, and measure accuracy gains from supplementary methods. We compare system performance on word relatedness and word association tasks, and it turns out that different methods have varying relative importance for these tasks..

Авторский указатель

Author's Index

Научное издание

Компьютерная лингвистика и интеллектуальные технологии

По материалам ежегодной
Международной конференции «Диалог»

Выпуск 14 (21). 2015

Том 2. Доклады специальных секций

Ответственный за выпуск **А. А. Белкина**
Вёрстка **К. А. Климентовский**

Подписано в печать 08.05.2015
Формат 152 × 235
Бумага офсетная
Тираж 250 экз. Заказ № 52

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