Компьютерная лингвистика и интеллектуальные технологии

По материалам ежегодной Международной конференции «Диалог» (2012)

Выпуск 11

В двух томах

Том 2. Доклады специальных секций

Computational Linguistics and Intellectual Technologies

Papers from the Annual International Conference “Dialogue” (2012)

Issue 11

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 № 12-06-06045-г

Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 30 мая–3 июня 2012 г.). Вып. 11 (18): В 2 т. Т. 2: Доклады специальных секций — М.: Изд-во РГГУ, 2012.

Сборник включает 78 докладов международной конференции по компьютерной лингвистике и интеллектуальным технологиям «Диалог 2012», 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

11-й выпуск ежегодника «Компьютерная лингвистика и интеллектуальные технологии» содержит материалы 18-й Международной конференции «Диалог». Для сборника было отобрано 78 докладов, охватывающих различные направления исследований в области компьютерного моделирования и анализа естественного языка. В настоящем сборнике представлены:

- Лингвистическая семантика и семантический анализ
- Формальные модели языка и их применение
- Теоретическая и компьютерная лексикография
- Методы оценки (evaluation) систем анализа текстов
- Корпусная лингвистика; создание, применение, оценка корпусов
- Интернет как лингвистический ресурс; лингвистические технологии в Интернете
- Извлечение знаний из текстов
- Компьютерный анализ документов: реферирование, классификация, поиск
- Автоматический анализ тональности текстов
- Модели синтаксиса для задачи синтаксического парсинга
- Машинный перевод
- Модели общения, коммуникация, диалог и речевой акт
- Анализ и синтез речи

«Диалог» – самая крупная конференция по компьютерной лингвистике, проводимая в России. Принципиальной особенностью конференции является пристальное внимание к технологиям автоматического анализа языка, основанным на лингвистических моделях. Именно этим объясняется и состав участников, и программа конференции, в которой соседствуют теоретические и прикладные исследования. На «Диалоге» представлены и работы, сделанные в рамках статистических подходов, и гибридные системы, что позволяет, в частности, сравнивать полученные результаты.

В этом году на «Диалоге» подводятся итоги двух тестовых испытаний: систем анализа тональности текстов (специальная дорожка РОМИПа) и систем синтаксического анализа (2-й этап Форума по тестированию систем автоматического анализа текстов). Выбор именно таких направлений тестирования оказался очень полезным, поскольку предложенные участникам задачи в этих двух испытаниях оказались принципиально разными с точки зрения методов оценки.

Тестирование анализа тональности представляет собой образцовую прикладную задачу компьютерной лингвистики с простыми и эффективными методами оценки. При анализе русскоязычных тестовых коллекций разработчики имели возможность использовать любые существующие или новые методы – и опубликованные результаты тестирования позволяют оценить их сравнительный потенциал.

Тестирование систем синтаксического анализа было направлено не на анализ какой-то частной прикладной задачи, а на качество полученной синтаксической разметки как таковое. Для русского языка подобное сравнительное тестирование проводилось впервые, и задача унификации используемых разработчиками синтаксических моделей оказалась весьма сложной.

Опубликованные в сборнике результаты показывают, с какими проблемами пришлось столкнуться как разработчикам, так и организаторам тестирования. Тем не менее этот новый шаг в разработке методов содержательного тестирования систем автоматического анализа русского языка представляется весьма поучительным и очень важным для «Диалога» с точки зрения его глобальной задачи.

В сборник включены итоговые статьи организаторов двух испытаний и большая часть комментирующих статей участников. Полностью статьи участников публикуются на сайте конференции и в её электронных материалах.

Постоянной доминантой «Диалога» является корпусная тематика. В прошлом году началась интенсивная дискуссия о методах оценки адекватности корпусов и применяемых методик корпусных исследований для решения различных задач лингвистики и лексикографии. Эта дискуссия имела продолжение после конференции и получила новое развитие в работах, представленных в этом году. В сборнике публикуются статьи, посвященные анализу практики применения существующих корпусов и проектам разработки корпусов нового типа.

Традиционно важное место в программе «Диалога», выросшего из междисциплинарных семинаров «Модели общения», занимают исследования звучащей речи и коммуникативных стратегий, в том числе невербальных.

Несмотря на широту тематики докладов этого года, они тем не менее не отражают полной картины направлений конференции. Более полное представление о проблематике «Диалога» можно получить на сайте конференции www.dialog-21.ru, где опубликованы обширные электронные архивы «Диалогов» прошлых лет.

Программный комитет конференции «Диалог»
Редакционная коллегия ежегодника «Компьютерная лингвистика и интеллектуальные технологии»

Организаторы

Ежегодная конференция «Диалог» проводится под патронажем Российского Фонда Фундаментальных Исследований при организационной поддержке компании ABBYY.

Основными учредителями конференции являются:

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

Конференция проводится при поддержке Российской ассоциации искусственного интеллекта.

Международный программный комитет

Буате Кристиан	Гренобльский университет
Богуславский Игорь Михайлович	Политехнический Университет Мадрида
Гельбух Александр Феликсович	Национальный политехнический институт, Мехико
Иомдин Леонид Лейбович	Институт проблем передачи информации РАН
Кибрик Александр Евгеньевич	Филологический факультет МГУ
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
Нивре 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
Сегалович Илья Валентинович	Компания Яndex
Селегей Владимир Павлович	Компания ABBYY
Хови Эдуард	University of Southern California
Шаров Сергей	University of Leeds, UK

Организационный комитет и Редсовет

Селегей Владимир Павлович, *председатель*	Компания ABBYY
Беликов Владимир Иванович	Институт Русского Языка им. В. В. Виноградова
Добров Борис Викторович	НИВЦ МГУ
Иомдин Леонид Лейбович	Институт проблем передачи информации РАН
Кобозева Ирина Михайловна	Филологический факультет МГУ
Козеренко Елена Борисовна, *руководитель проекта по гранту РФФИ 12-06-06045-г*	Институт проблем информатики РАН
Лауфер Наталия Исаевна	ООО «проФан Продакшн»
Ляшевская Ольга Николаевна	Universitetet i Tromsø, Norway
Соколова Елена Григорьевна	Институт лингвистики, РГГУ
Толдова Светлана Юрьевна	Филологический факультет МГУ
Шаров Сергей	University of Leeds, UK

Секретариат

Талис Валентина Львовна, *секретарь оргкомитета, редактор сайта*	Компания ABBYY
Мытникова Татьяна Александровна, *координатор*	Компания ABBYY
Морозова Юлия Игоревна	ИПИ РАН

Рецензенты

Августинова Таня
Азарова Ирина Владимировна
Апресян Валентина Юрьевна
Баранов Анатолий Николаевич
Беликов Владимир Иванович
Богданова Наталья Викторовна
Богданов Алексей Владимирович
Бонч-Осмоловская Анастасия
Браславский Павел Исаакович
Гельбух Александр Феликсович
Горностай Татьяна Александровна
Губин Максим Вадимович
Даниэль Михаил Александрович
Добров Борис Викторович
Добровольский Дмитрий Олегович
Добрынин Владимир Юрьевич
Зарецкая Елена Наумовна
Захаров Леонид Михайлович
Зуев Константин Алексеевич
Иодин Борис Леонидович
Иомдин Леонид Лейбович
Кибрик Андрей Александрович
Кобозева Ирина Михайловна
Козеренко Елена Борисовна
Крейдлин Григорий Ефимович
Кронгауз Максим Анисимович
Лапшин Владимир Анатольевич
Лахути Делир Гасемович
Левонтина Ирина Борисовна
Лобанов Борис Мефодьевич
Лукашевич Наталья Валентиновна
Ляшевская Ольга Николаевна
Маккарти Диана
Новицкий Валерий Игоревич
Нивре Йоахим
Пазельская Анна Германовна
Плунгян Владимир Александрович
Раскин Виктор
Рахилина Екатерина Владимировна
Савельев Василий Евгеньевич
Селегей Владимир Павлович
Сокирко Алексей Викторович
Соколова Елена Григорьевна
Тестелец Яков Георгиевич
Тихомиров Илья Александрович
Урысон Елена Владимировна
Филиппова Екатерина Александровна
Хорошевский Владимир Федорович
Циммерлинг Антон Владимирович
Хови Эдуард
Шаров Сергей Александрович
Юдина Мария Владимировна
Янко Татьяна Евгеньевна

Содержание*

Раздел II.
Доклады, представленные участниками тестирования систем анализа тональности

* Доклады внутри каждого раздела упорядочены по фамилии первого автора в соответствии с порядком английского алфавита.

Раздел III.
Доклады, представленные участниками тестирования систем синтаксического анализа

Раздел II.
Доклады, представленные участниками тестирования систем анализа тональности

В данном разделе публикуется итоговая статья организаторов тестирования систем анализа тональности и отдельные статьи участников тестирования. Полностью с комментирующими сообщениями участников можно ознакомиться на сайте конференции «Диалог».

SENTIMENT ANALYSIS TRACK AT ROMIP 2011

Chetviorkin I. I. (ilia2010@yandex.ru)
Lomonosov Moscow State University

Braslavski P. I. (pbraslavski@acm.org)
Kontur Labs, Ural Federal University

Loukachevitch N. V. (louk_nat@mail.ru)
Research Computing Center of Lomonosov
Moscow State University

Russian Information Retrieval Seminar (ROMIP) is a Russian TREC-like IR evaluation initiative. In 2011 ROMIP launched a new track on sentiment analysis. Within the track we prepared a training collection of user reviews along with ratings for movies, books, and digital cameras. Additionally, we compiled a test collection of blog posts with reviews in the same domains and labeled them according to expressed sentiment. The paper describes the collections' characteristics, track tasks, the labeling process, and evaluation metrics. We summarize the participants' results and make suggestion for future editions of the track.

Key words: ROMIP, sentiment classification, sentiment analysis, opinion mining, blog data

1. Introduction

With the development of internet technologies an increasingly large number of people have got an opportunity to express their opinions on the web. Journal-like web pages (weblogs) allows internet users to share their feelings, emotions and attitudes about various products, services, and real-life events with other people. This information can be very useful both for other web users and for service providers or product manufacturers.

Extremely accessible blog software has facilitated blogging for a wide audience, and, as a result, boosted the growth rate of information available online. Thus, the blogosphere has become a highly dynamic subset of the World Wide Web that evolves responding to real-world events and offers several new research areas.

Today, sentiment analysis research attracts a lot of interest as a tool for opinion processing and company reputation management. Sentiment analysis has a lot of different subtasks [Pang&Lee2008]. The most well-known of them are:

- subjectivity/objectivity identification;
- polarity classification of a given text at the document, sentence, or feature/aspect level;
- advanced, "beyond polarity", sentiment classification that looks, for instance, at emotional states such as "angry," "sad," and "happy";
- recognition of sarcastic sentences (phrases);
- feature/aspect-based sentiment analysis;
- sentiment summarization.

Russian Information Retrieval Seminar (ROMIP, http://romip.ru) is a Russian TREC-like information retrieval evaluation initiative. It was launched in 2002 to increase communication and support research community (both academia and industry) in the area of IR in Russian by providing a basis for independent evaluation of IR methods. Since its start, ROMIP has organized a number of different tracks, e. g. ad hoc retrieval, snippet generation, document classification, question answering (QA), and image retrieval. ROMIP prepared and made available for researchers a number of data collections.

In many respects ROMIP seminars are similar to other international information retrieval events such as TREC and NTCIR, which have already conducted different sentiment analysis tracks (see Section 2). We decided to start with sentiment classification of reviews in Russian because it was quite simple to find data, but the good quality of classification was rather difficult to achieve. On the other hand, we were interested in the state of the art in this research area.

The task of the ROMIP 2011 sentiment analysis track was to classify blog posts about different products according to sentiment expressed in documents. It was reported in the literature that the more classes there are, the harder it is to classify a text by sentiment. Thus, in the first pilot run of the track in 2011 we had three tasks:

- two-class classification task,
- three-class classification task,
- five-class classification task.

It was the first shared task evaluation of document sentiment classification in Russian.

The rest of this paper is structured as follows. In Section 2, we make a brief overview of similar evaluation campaigns and available datasets. Section 3 provides a short description of the newly created collections used for training and evaluation. Section 4 describes the sentiment classification task. Section 5 provides an overview of runs the submitted by participants. Concluding remarks can be found in Section 6.

2. Related evaluation campaigns and datasets

In this section we briefly overview cognate evaluation campaigns within TREC (http://trec.nist.gov) and NTCIR (http://research.nii.ac.jp/ntcir/index-en.html), as well as provide a list of datasets available for research. [Pang&Lee2008] gives a good overview of evaluation initiatives, available data and resources in opinion mining and sentiment analysis. However, some new datasets and shared tasks emerged after the book had been published.

2.1. TREC

Blog track was organized in 2006–2010 within TREC initiative [Macdonald2010, Ounis2008]. In 2006–2008 the track investigated an opinion-finding task, complemented with a polarity subtask in 2007–2008.

In the opinion-finding task, participating systems had to retrieve opinionated posts about a given target such as person, location or organization, concept (such as type of technology), product name or event. Both *relevance* and *opinionatedness* of retrieved posts were judged. Additionally, polarity of the opinion expressed in relevant posts was labeled as *positive*, *negative*, or *mixed*. This labeling led to a supplemental polarity subtask in two subsequent years. In 2007 the task was formulated as a classification task, i. e. for each retrieved post participants should have predicted its polarity. For TREC 2008, this task was reformulated as a ranking task: only posts expressing polarity should have been retrieved and ranked by the degree of positivity or negativity respectively.

The aforementioned experiments within TREC were performed on the TREC Blogs06 collection. Blogs06 is a collection of over 3.2 million permalinks (i. e. a single blog post and all associated comments) from over 100,000 blogs that had been crawled during an 11-week period from 6th December 2005 until 21st February 2006. To make settings more realistic, a sample of spam blogs, news feeds, as well as non-English documents was injected. (This collection was also used within TAC 2008 Opinion QA Task, http://www.nist.gov/tac/data/past/2008/OpSummQA08.html)

URL: http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG

2.2. NTCIR

NTCIR, a Japanese counterpart of TREC, launched a pilot opinion track in 2006. The dataset was compiled from news articles in Japanese, Chinese, and English. Participants had to solve the following tasks on the sentence level: 1) detection of opinionated sentences, 2) detection of opinion holders, 3) sentence relevance to the topic, and 4) polarity labeling as *positive*, *negative*, or *neutral* [Seki2007]. In NTCR-7 the track evolved into Multilingual Opinion Analysis Track (MOAT); documents in Simplified Chinese and the opinion target identification subtask were added. Moreover, some tasks were performed with finer granularity, i. e. identification was applied to sentence fragments [Seki2008]. In NTCIR-8 the subtasks were extended towards cross-language analysis and question answering: opinionated answers in different languages had to be extracted in response to questions in English [Seki2010].

URL: http://research.nii.ac.jp/ntcir/permission/ntcir-6/perm-en-OPINION.html
http://research.nii.ac.jp/ntcir/permission/ntcir-7/perm-en-MOAT.html

2.3. Data collections

What follows is a non-exhaustive list of datasets not associated with established evaluation campaigns, which can be used for sentiment and opinion analysis. Some of the datasets are no longer available and are mentioned here for reference only. The terms and conditions under which the data are released may vary, so please consult provided URLs.

Cornell Movie Review Datasets contains reviews from IMDb (http://imdb.com). There are 1,000 'polarity reviews' tagged positive or negative, as well as a larger amount of original reviews along with users' star ratings.

URL: http://www.cs.cornell.edu/people/pabo/movie-review-data/

Bing Liu and colleagues compiled several dataset and made them available for researchers in opinion mining and sentiment analysis. The most notable is probably the **Amazon Product Review Dataset** containing 5.8M+ reviews on books, music, DVDs and consumer electronics.

URL: http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

One of the first sizeable blog datasets available for research was **BlogPulse 2005 dataset** released to participants in the Workshop on Weblogging Ecosystem (WWE) in 2006. The dataset contained 10M posts from 1M weblogs collected during three weeks in July 2005.

URL (as preserved in Web Archive): http://web.archive.org/web/20090615025713/http://www.blogpulse.com/www2006-workshop/cfp.html

Several datasets were made available through International Conference on Weblogs and Social Media (http://www.icwsm.org), which continued the tradition from the WWE2006 workshop.

Nielsen BuzzMetrics 2006 Dataset contains 14M weblog posts in XML format from 3M weblogs published in May 2006. The dataset contains posts in different languages (e. g. about 6 % of posts are reported to be in Russian).

URL: http://www.icwsm.org/data.html

In the following years much bigger datasets were compiled and released. **ICWSM 2009 Spinn3r Blog Dataset** contains posts made between August 1st and October 1st, 2008 along with some metadata, 44 million blog posts in total. **ICWSM 2011 Spinn3r Dataset** is one magnitude bigger and much more versatile — it covers blog posts, news articles, classifieds, forum posts, and social media content created between January 13th and February 14th 2011, resulting in 386 million items.

URL: http://www.icwsm.org/data/

Content Analysis in Web 2.0 (CAW 2.0) is a dataset associated with a workshop of the same name at the WWW2009 conference. The dataset comprises tweets, forum discussions, comments on news, movie reviews, and on-line chats that total to 680K messages. Workshop organizers offered a number of shared tasks on these data, including opinion and sentiment analysis. The sentiment analysis task was to assign a message to categories *neutral*, *happy*, *angry* or *sad* (fuzzy assignments were allowed); whereas opinion tasks dealt with three categories: *factual*, *opinionated-positive* and *opinionated-negative*.

URL: http://caw2.barcelonamedia.org/node/7

The main task of the **TREC Microblog** track is *ad hoc* retrieval in tweets. However, we envision that the track data collection — 16 million tweets sampled between January 23rd and February 8th, 2011 — might be employed for sentiment analysis and opinion mining research.

URL: http://trec.nist.gov/data/tweets/

CyberEmotions is an integrating, ongoing, large-scale European research project focusing on the role of collective emotions in creating, forming and breaking-up eCommunities. One of the project outcomes is the creation of a corpus that consists of three parts: 1) 2,5M+ comments from BBC News forum, including 1K+ labeled items; 2) Digg post comments (1.6M+ comments, including 1K+ labeled items); and 3) MySpace comments exchanged between pairs of friends from a total of 100K+ social network members (including 1K+ labeled items).

URL: http://www.cyberemotions.eu/data.html

The **MPQA Opinion Corpus** contains news articles from a wide variety of news sources manually annotated for opinions and other private states (i. e. beliefs, emotions, sentiments, speculations, etc.).

URL: http://www.cs.pitt.edu/mpqa/

The **Multi-Domain Sentiment Dataset** consists of product reviews taken from Amazon.com with many product types (domains). Some domains (books and DVDs) have hundreds of thousands of reviews. Others (musical instruments) have only a few hundred.

URL: http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

3. ROMIP Data Collections

For the sentiment classification tasks we chose three different domains: movies, books, and digital cameras. Movie and book collections (15,718 and 24,159 reviews, respectively) were obtained from online recommendation service IMHONET (http://www.imhonet.ru). Each review in these collections had user's score on a ten-point scale (zero

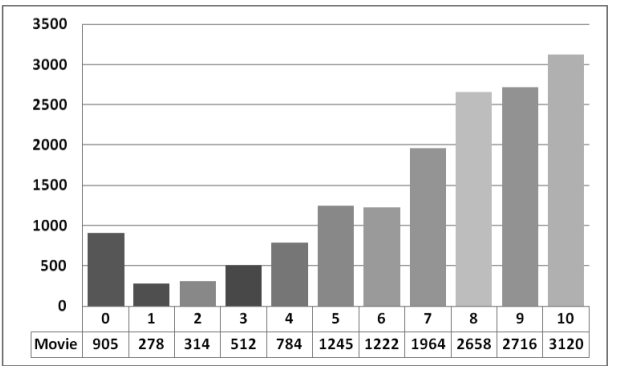
means unmarked). The digital camera review collection (10,370 reviews) was provided by Yandex. Reviews for cameras were collected from the Yandex.Market comparison shopping service (http://market.yandex.ru) and had users' scores on a five-point scale.

The average review length in the movie domain was 72 words, 49 words in the book domain, and 101 words in the camera domain. Score distributions can be found in Fig. 1–3.

These three collections were presented to participants for training their algorithms. No additional information was provided.

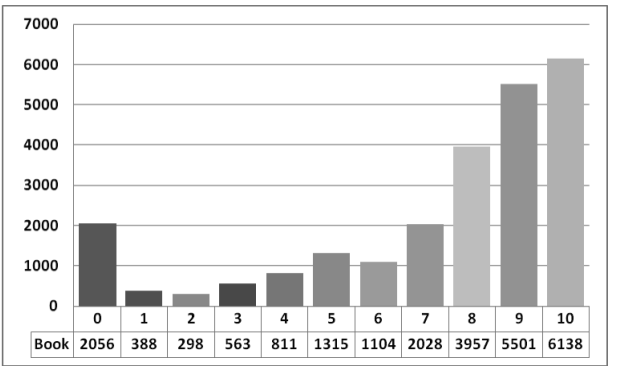
To evaluate the quality of sentiment classification algorithms, we needed additional collections without any authors' scores. We decided to collect blog posts about various entities in three domains. For this purpose we used Yandex's Blog Search Engine (http://blog.yandex.ru).

For each domain a list of search queries was manually compiled. There were 61 book queries, 922 camera queries, and 112 movie queries. Each query was about only one entity (or related objects) from selected domains. There is a query example from the book domain: [vpechatlenyia ot kniga “Victor Pelevin” -spisok] *[impression from the book “Victor Pelevin” -list].*



	0	1	2	3	4	5	6	7	8	9	10
Movie	905	278	314	512	784	1245	1222	1964	2658	2716	3120

Figure 1. Score distribution in movie review collection



	0	1	2	3	4	5	6	7	8	9	10
Book	2056	388	298	563	811	1315	1104	2028	3957	5501	6138

Figure 2. Score distribution in book review collection

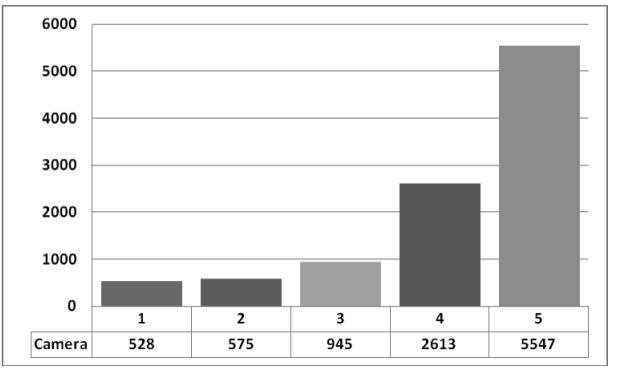


Figure 3. Score distribution in camera review collection

For each query we obtained a set of blog posts (both relevant and irrelevant). Finally results for all queries were merged. The resulting collection included 16,821 reviews for entities from various domains. The average review length in this collection was 1,146 words. Participating systems had to return sentiment labels for all these documents.

4. Assessment Procedure

Test collection included a lot of irrelevant texts, reviews containing sentiment about various topics or texts with both subjective and objective information. Since we wanted to solve only document sentiment classification task we had to select for evaluation only strongly subjective texts with one dominant topic related to entities in the target domains. As a result we selected 275 book reviews, 329 movie reviews, and 270 digital camera reviews for testing.

At the next step, all reviews were labeled by two assessors with three scores (at once) on different scales ***S***:

- ***S*** = ***{1, 2}*** for two-class classification task, where 1 — a negative review and 2 — a positive review;
- ***S*** = ***{1, 2, 3}*** for-three class classification task, where 1 — a generally negative review, 2 — a review has significant positive and negative aspects of the evaluated entity, 3 — a generally positive review;
- ***S*** = ***{1, 2, 3, 4, 5}*** for five-class classification task, where 1 — a generally negative review, 2 — a generally negative, but points to some positive aspects of the entity, 3 — a review has significant positive and negative aspects of the evaluated entity, 4 — a generally positive, but points to some negative aspects of the entity, 5 — a generally positive review.

Class distribution for each task was highly skewed. For example, in the two-class task we had 84% of positive reviews for cameras, 92% of positive reviews for books and 85% of positive reviews for movies. In the three-class and the five-class tasks we had the same situation — the majority of reviews were positive.

In Table 1 one can find Cohen's kappa coefficient for measuring the inter-rater agreement.

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

where Pr(a) is the relative observed agreement among raters, and Pr(e) is the hypothetical probability of chance agreement.

Table 1. Kappa coefficients for different tasks

Kappa	2 classes	3 classes	5 classes
Movies	0.818	0.615	0.429
Books	0.812	0.674	0.545
Digital Cameras	0.808	0.602	0.398

Proportion of reviews that were assigned the same score by both assessors for each task-domain pair can be found in Table 2.

Table 2. Proportion of reviews in AND evaluation scheme

	2 classes	3 classes	5 classes
Movies	0.948	0.799	0.590
Books	0.967	0.829	0.684
Digital Cameras	0.944	0.766	0.548

5. Results Overview

In all, twelve groups took part in the sentiment classification task. There were 105 submitted runs in the two-class task, 81 runs in the three-class task, and 30 runs in the five-class task. We used different metrics to evaluate the quality of classification algorithms.

5.1. Official metrics

The metrics used for the opinion classification task were *precision, recall, F1-measure, accuracy and average Euclidian distance*. For the first three measures we used traditional (separately for each category) and macro-averaged variants.

Macro metrics show classification quality for all classes, while traditional metrics evaluate the quality of algorithms only in relation to one specific class. Macro metrics are convenient for multiclass classification tasks to account for imbalanced test data. Since we had highly imbalanced test collection (see Section 4) we used

macro-averaged metrics to evaluate the ability of algorithms to determine each of the classes.

To give definition to all these metrics, we assume that:

- $\mathbf{tp_x}$ is the number of objects correctly classified as class X by the algorithm,
- $\mathbf{fp_x}$ is the number of objects falsely classified as class X,
- $\mathbf{fn_x}$ is the number of objects belonging to class X, but classified as non-X by the algorithm,
- $\mathbf{tn_x}$ the number of objects classified to non-X and they actually belong to one of the non-X classes

Table 3. Classifier output types

	actual class	
predicted class	$\mathbf{tp_x}$ (true positive) Correct result	$\mathbf{fp_x}$ (false positive) Unexpected result
	$\mathbf{fn_x}$ (false negative) Missing result	$\mathbf{tn_x}$ (true negative) Correct absence of result

Precision is the proportion of objects classified as X that truly belong to class X. The macro variant of this feature averages all class precision values.

$$P = \frac{tp_x}{tp_x + fp_x}$$

$$Macro\_P = \frac{1}{|S|} \cdot \sum_{x \in S} \frac{tp_x}{tp_x + fp_x}$$

Recall is the proportion of all objects of class X that is classified by the algorithm as X. The macro variant of this feature averages all class recall values.

$$R = \frac{tp_x}{tp_x + fn_x}$$

$$Macro\_R = \frac{1}{|S|} \cdot \sum_{x \in S} \frac{tp_x}{tp_x + fn_x}$$

F1-measure is the harmonic mean of Precision and Recall. Macro_F1 is the average from all F1-measures of particular classes.

$$Fmeasure = \frac{2 \cdot P \cdot R}{P + R}$$

Accuracy is proportion of correctly classified objects in all objects processed by the algorithm.

$$Accuracy = \frac{tp_x + tn_x}{tp_x + tn_x + fp_x + fn_x}$$

Average Euclidean distance is the average from the quadratic difference between the scores of the algorithm and the assessor scores (average of the assessors' scores).

$$D = \sqrt{\frac{\sum_{i=1}^{n}(q_i - p_i)^2}{n}}$$

5.2. Participants' results

For each task we calculated baseline values for all measures. We took as the baseline a dummy classifier that assigns all reviews to the most frequent class. For this reason, the maximum value for all macro metrics was equal to one divided by the number of classes in the task, which was rather low in comparison with participants' runs. On the other hand, the accuracy and average Euclidian distance were very close to the best results.

In addition, two evaluation schemes were applied:

- **AND**, only those reviews that have the same score from both assessors were involved in evaluation (see Section 4)
- **OR,** we considered an answer of the algorithm to be the right one if it matched with the answer of at least one assessor

In addition, it was important to determine if the difference (according to task's primary measures) between the best runs was statistically significant. For this purpose the Wilcoxon signed-rank test/Two-tailed test ($\alpha = 0.05$) was used. We marked the top result with "*" in case of insignificant difference with the second result.

Two-class task

Primary measures for evaluating the two-class classification performance were macro-F1 and accuracy. Table 4 shows the best two runs for each type of entities for evaluation scheme OR in terms of macro F1-measure and accuracy. Table 5 shows similar results for evaluation scheme AND.

Table 4. Two-class classification results (OR)

Run_ID	*Object*	*Macro_P*	*Macro_R*	*Macro_F1*	*Accuracy*
xxx-40	book	0.714	0.804	**0.747**	0.895
xxx-0	book	0.751	0.721	0.735	0.924
xxx-24 (46)	book	0.968	0.630	0.690	**0.938***
xxx-19	book	0.790	0.651	0.694	0.931
Baseline	book	0.460	0.500	0.479	0.920

Run_ID	*Object*	*Macro_P*	*Macro_R*	*Macro_F1*	*Accuracy*
yyy-24	camera	0.918	0.940	**0.929***	**0.959***
yyy-16	camera	0.944	0.898	0.919	0.956
Baseline	camera	0.426	0.500	0.460	0.852
zzz-23	film	0.776	0.797	**0.786**	**0.881**
zzz-9	film	0.706	0.794	0.730	0.812
zzz-14	film	0.743	0.597	0.623	0.860
Baseline	film	0.427	0.500	0.461	0.854

Table 5. Two-class classification results (AND)

Run_ID	*Object*	*Macro_P*	*Macro_R*	*Macro_F1*	*Accuracy*
xxx-34	book	0.698	0.761	**0.723**	0.902
xxx-0	book	0.739	0.709	0.723	0.921
xxx-24 (46)	book	0.967	0.614	0.668	**0.936***
xxx-19	book	0.789	0.651	0.693	0.929
Baseline	book	0.459	0.500	0.478	0.917
yyy-24	camera	0.909	0.934	**0.921***	**0.957***
yyy-16	camera	0.936	0.881	0.905	0.953
yyy-9	camera	0.890	0.929	0.908	0.949
Baseline	camera	0.422	0.500	0.457	0.843
zzz-23	film	0.760	0.781	**0.770**	**0.875**
zzz-9	film	0.680	0.772	0.702	0.801
zzz-14	film	0.715	0.580	0.600	0.853
Baseline	film	0.423	0.500	0.458	0.846

Results in these two evaluation scheme s are highly correlated. For schema AND, the results are slightly worse, because all reviews with ambiguous scores were excluded (any algorithm answer was correct in the OR scheme). For the three-class and the five-class tasks we give results only for OR.

According to the results, reviews in different domains have different complexity. Traditionally, [Turney2002] the movie domain is the most difficult one (in accordance with accuracy).

All best runs have outperformed the baseline, but not all participants did.

Three-class task

In this task, primary measures were the same as in the previous task: macro F1-measure and accuracy. Table 6 shows the two best results for each object. The results and baselines drop significantly in comparison with the two-class task.

Table 6. Three-class classification results (OR)

Run_ID	*Object*	*Macro_P*	*Macro_R*	*Macro_F1*	*Accuracy*
xxx-3	book	0.677	0.532	**0.577***	0.756
xxx-43	book	0.671	0.517	0.570	0.756
xxx-11	book	0.658	0.475	0.488	**0.771**
xxx-36	book	0.625	0.481	0.499	0.764
Baseline	book	0.227	0.333	0.270	0.68
yyy-3	camera	0.843	0.594	**0.663***	**0.841***
yyy-11	camera	0.797	0.596	0.661	0.815
Baseline	camera	0.216	0.333	0.262	0.648
zzz-10	film	0.671	0.535	**0.592***	**0.754***
zzz-1	film	0.661	0.524	0.584	0.751
zzz-19	film	0.657	0.526	0.582	0.754
Baseline	film	0.235	0.333	0.276	0.705

Classifying camera reviews seems to be easier than classifying reviews from the other domains.

Five-class task

The five-class classification task differs significantly from previous tasks. Even though such evaluation scheme is very common on the internet (“five stars” system), it is a quite difficult task because not only does one need to determine the text’s sentiment, but it is also necessary to find its strength (rating-inference problem). Even assessors' agreement in five-class labeling is much lower than it is in other tasks.

Accuracy and average Euclidian distance were the primary measures for this task. Firstly, it was important to know what percentage of reviews was classified correctly, secondly, what was the average score deviation from assessors' scores.

Table 7. Five-class classification results (OR)

Run_ID	*Object*	*Avg_Eucl_Distance*	*Macro_F1*	*Accuracy*
xxx-7	book	**0.872***	0.284	**0.622***
xxx-4 (9)	book	0.892	**0.291***	0.622
xxx-5	book	0.972	0.270	0.615
Baseline	book	0.909	0.123	0.48
yyy-1	camera	**0.928**	0.298	0.567
yyy-3	camera	0.940	0.287	0.570
yyy-4	camera	0.971	**0.342**	**0.626**
yyy-2	camera	1.215	0.332	0.626

Run_ID	*Object*	*Avg_Eucl_Distance*	*Macro_F1*	*Accuracy*
Baseline	camera	1.165	0.144	0.563
zzz-1 (5)	film	**1.026***	**0.286***	0.599
zzz-2	film	1.071	0.266	0.559
zzz-6	film	1.133	0.247	**0.602**
Baseline	film	1.460	0.135	0.506

In all domains F1-measure is very low. In comparison to the accuracy level it means that it is difficult for the algorithms to classify reviews from minority classes.

6. Conclusions

ROMIP 2011 was the first shared task evaluation of text sentiment classification in Russian. New collections in different domains (movies, books, digital cameras) were created and made available for research. We thought that sentiment classification was rather a challenging task and it was important to know the state of art for Russian language.

In each task/domain pair the best runs show quite high performance despite highly unbalanced test collection. Based on these results we can conclude that each domain has different complexity and each of them requires an additional adaptation of the algorithms.

We discovered that the interest in sentiment analysis of Russian texts was very high among researchers and specialists in natural language processing. Results in each task coincide with the results for other languages described in literature. At ROMIP 2012 we are planning to offer two new tasks: subjectivity\objectivity identification task and detection of review's domain.

Instructions of how to obtain any of ROMIP collections can be found at http://romip.ru/ru/participation.

Acknowledgements. We are grateful to Yandex and IMHONET for granting their review data collections for research purposes of the seminar. We thank Marina Nekrestyanova, Maxim Gubin and Boris Dobrov for many valuable comments and help with ROMIP organization. We also thank Alya Ageeva for proofreading the paper. This work is partially supported by RFBR grant N11-07-00588-a.

References

1. *Macdonald C., Santos R. L., Ounis I., and Soboroff I.* (2010) Blog track research at TREC. SIGIR Forum 44(1), pp 58–75.
2. *Ounis I., Macdonald C. and Soboroff I.* (2008) On the TREC Blog Track. In Proceedings of International Conference on Weblogs and Social Media (ICWSM 2008).

3. *Pang B., Lee L.*: Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval. Now Publishers, 2008.
4. *Seki Y., Evans D. K., Ku L. W., Chen H. H., Kando N. and Lin C. Y.* (2007) Overview of Opinion Analysis Pilot Task at NTCIR-6. In Proc. of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies, pp 265–278.
5. *Seki Y., Evans D. K., Ku L. W., Sun L., Chen H. H. and Kando N.* (2008) Overview of Multilingual Opinion Analysis Task at NTCIR-7. In Proc. of the Seventh NTCIR Workshop Meeting on Evaluation of Information Access Technologies, pp 185–203.
6. *Seki Y., Ku L. W., Sun L., Chen H. H. and Kando N.* (2010) Overview of Multilingual Opinion Analysis Task at NTCIR-8 — A Step Toward Cross Lingual Opinion Analysis. In Proc. of the Eights NTCIR Workshop Meeting on Evaluation of Information Access Technologies, pp 209–220
7. *Turney P. D.* (2002) Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Procs. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02). pp. 417–424.

TESTING THE SENTIMENT CLASSIFICATION APPROACH IN VARIOUS DOMAINS — ROMIP 2011

Chetviorkin I. I. (ilia2010@yandex.ru)

Faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University

We offer a review of sentiment classification experiments in various domains using different training sets. In the movie domain we studied the impact of opinion word weights on the quality of classification. We selected the best feature set and ran them on each task-domain pair. In several tasks our algorithm achieved high quality of the classification.

Key words: ROMIP, sentiment classification, opinion words, domain adaptation

1. Introduction

This year within Russian Information Retrieval Seminar a new sentiment analysis track was offered to the participants. This track had three tasks related to the classification of documents by sentiment expressed in them:

- two-class classification task,
- three-class classification task,
- five-class classification task.

In addition the documents (blog posts) from the test collection were about entities from various domains: books, movies and digital cameras. Each domain requires extra tuning of the algorithms and it can be difficult to achieve a good performance in all domains.

The easiest task is to classify reviews into two classes: *positive* and *negative* [Pang and Lee, 2008]. Quality of two-way classification using the topic-based categorization approach for reviews exceeds 80% [Pang et al., 2002]. In [Whitelaw et al., 2005] the quality of review classification, based on the so-called appraisal taxonomy, is described as 90.2%.

However, when we turn to the problem of review division into three classes, the quality of automatic classification decreases to 75% after an adjustment to an individual author's style, and 66.3% in a case of author independent test collection [Pang and Lee, 2005].

In rating-inference problem with four classes reported accuracy is 54.6% using metric labeling formulation [Pang and Lee, 2005] and 59.2% using graph-based

semi-supervised learning algorithm with adjustment to an author style [Goldberg et al., 2006].

Recently we had conducted the similar research for the three-way classification problem in the movie domain [Chetviorkin and Loukachevitch, 2011a]. It was interesting to compare our results with other participants and to try to utilize our approach in the two-class and five-class tasks in various domains.

In the current paper we describe our classification approach using such features as word weights, opinion words and polarity influencers. We have submitted five runs for the three-way classification task in the movie domain and one run (with complete set of features) for all other combinations of tasks and domains.

The reminder of this paper is structured as follows. Section 2 provides a short description of the training collections. Section 3 briefly describes our approach to the sentiment classification. Section 4 gives an overview of our submission results. We provide concluding remarks in Section 5.

2. Data Collections

All participants were granted three train collections, one for each domain (for score distribution in these collections see [Chetviorkin et al., 2012]). But we had created our own collections from the same sources earlier. It was more convenient for us to use our collections in the experiments.

Our movie and book collections (28,773 and 15,113 reviews accordingly) were collected from the online recommendation service *www.imhonet.ru*. Each review in these collections had user's score on a ten-point scale. The digital camera review collection (8,181 reviews) was collected from the Yandex.Market service and had user's score on a five-point scale. Score distributions in there three collections can be found in Fig.1–3.

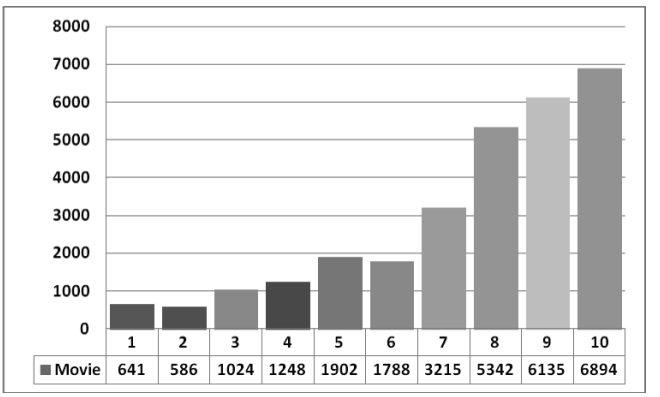


Figure 1. Score distribution in the movie review collection

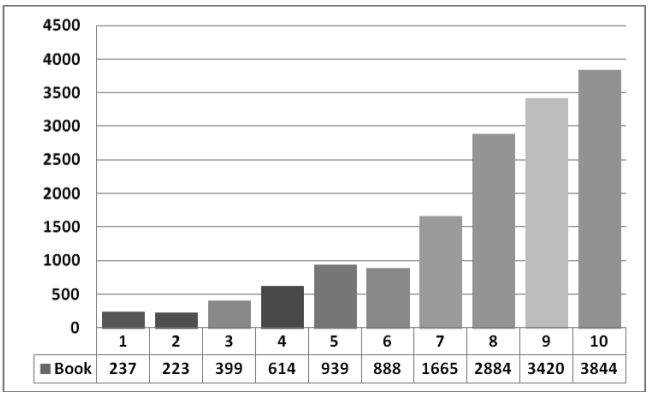


Figure 2. Score distribution in the book review collection

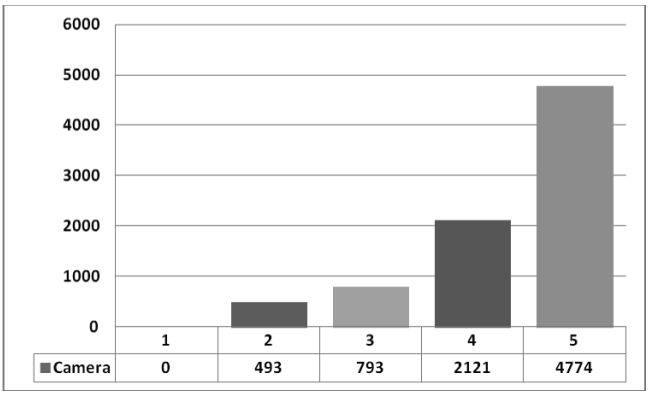


Figure 3. Score distribution in the camera review collection

In addition all participants gained the test collection with 16,821 blog posts about various entities.

3. Sentiment Classification Algorithm

In the sentiment classification track we used the same approach as provided in [Chetviorkin and Loukachevitch, 2011a]. We will shortly describe the main points of our algorithm and major changes, which were applied to it in correspondence with the various tasks and domains.

3.1. Features for review classification

In this research we utilized the best feature combinations which were obtained during the three-way classification experiments in the movie domain [Chetviorkin

and Loukachevitch, 2011a]. To improve the quality of the review classification we analyzed the following features:

- word weights based on different collections,
- opinion words,
- use of polarity influencers: they may reverse or enhance (*not, very*) polarity of other words,
- length and structure of reviews,
- use of punctuation marks

The best results were achieved using the bag of words (all words from the train collection with frequencies higher than four), TFIDF word weights, polarity influencers and opinion word weights.

TFIDF

The main elements of our feature set were lemmas, which appeared in the train collection more than three times. The simplest approach for document classification was to create feature vectors using binary weights of words, but not the most effective.

To improve the quality of classification we used TFIDF weights [Ageev et al., 2004] for lemmas with inversed document frequency calculated using the news collection with one million documents.

$$TFIDF(l) = \beta + (1-\beta) \cdot tf(l) \cdot idf(l)$$

$$tf_D(l) = \frac{freq_D(l)}{freq_D(l) + 0.5 + 1.5 \cdot \frac{dl_D}{avg\_dl}} \qquad idf(l) = \frac{\log(\frac{|c| + 0.5}{df(l)})}{\log(|c| + 1)}$$

- $freq_D(l)$ — number of occurrences of l in a document D,
- $dl_D(l)$ — length measure of a document D, in our case, it is number of terms in a review,
- avg_dl — average length of a document,
- $df(l)$ — number of documents in a collection (e. g. description or news collection) where term l appears,
- $\beta = 0.4$,
- $|c|$ — total number of documents in a collection.

Opinion words

Opinion words are the main polarity carriers in a text. We tried to utilize them in various ways in a combination with a bag of words [Chetviorkin and Loukachevitch, 2011a]. Only one useful variant was found: to modify word weights accordingly to opinion word weights in the extraction model.

We used our algorithm [Chetviorkin and Loukachevitch, 2011b] to extract high quality domain dependent opinion words. To generate the list of such words, four text collections were exploited: the review collection about entities from a specific domain, the collection of entity descriptions, the special small corpus and the collection of general news. On the basis of these collections a set of statistical features for words mentioned in reviews was calculated. We trained our model using word feature vectors in the movie domain and then utilized this model in two other domains. As a result we obtained a list of sentiment words for each domain, ordered by the predicted probability of their opinion orientation (opinion weight).

There are examples of opinion words with high probability value in the movie domain:

- *Trogatel'nyi (affecting), otstoi (trash), fignia (crap), otvratitel'no (disgustingly), posredstvenniy (satisfactory), predskazuemyi (predictable), ljubimyj (love) etc.*

In the review classification tasks we modified the weight of each word in the feature vectors as follows:

$$wordweight(x) = TFIDF(x) \cdot e^{opinweight(x)-0.5}$$

Thus, we increased weights of words with high opinion weight, and decreased weights of other words.

Polarity influencers

We used the same set of polarity influencers in all domains:

- operator (–): *net (no), ne (not)*;
- operator (+): *polnyj (full), ochen' (very), sil'no (strongly), takoj (such), prosto (simply), absoljutno (absolutely), nastol'ko (so), samyj (the most)*.

On the basis of this polarity shifter list we substituted sequences "polarity influencer word" using special operator symbols ("+" or "–") depending on an polarity shifter, for example:

NE HOROSHIJ (NOT GOOD) → –HOROSHIJ (— GOOD)
SAMYJ KRASIVYJ (THE MOST BEAUTIFUL) → + KRASIVYJ (+ BEAUTIFUL)
NASTOL'KO KRASIVYJ (SO BEAUTIFUL) → + KRASIVYJ (+ BEAUTIFUL)

Thus we added to the review vector representation only the operator phrases but not both words. It allowed us to take into account the impact of the polarity influencers.

3.2. Classification algorithm

Authors of previous studies almost unanimously agreed that Support Vector Machine algorithm works better for text classification tasks (and review classification

in particular) [Pang and Lee, 2008]. In view of the fact that we had a large amount of data and features (bag of words), library LIBLINEAR was chosen [Fan et al., 2008]. All parameters of the algorithm were left in accordance with their default values.

3.3. Scale mapping

To train our algorithm for classification in a certain scale, we need to map scores from the train collection scale to the task scale. We used the following mapping functions:

- **Two-class task:** {1–7} → "1" (thumbs down), {8–10} → "3" (thumbs up)
- **Three-class task:** {1–6} → "1" (thumbs down), {7–8} → "2" (so-so), {9–10} → "3" (thumbs up)
- **Five-class task:** {1–3} → "1", {4–5} → "2", {6–7} → "3", {8} → "4", {9–10} → "5"

For the digital camera collection we firstly multiplied each user's score by two and then used aforementioned mapping schemes.

It is rather important to choose a correct mapping function. We investigated the best mapping functions for the three-way classification problem in previous studies [Loukachevitch and Chetviorkin, 2011]. For the two other tasks we used our insights to define the mapping functions.

4. Results Overview

We have submitted five runs for the three-class task in the movie domain:

- Bag of words with TFIDF word weights (**BoW+tfidf**)
- Bag of words with opinion word weights (**BoW+opweight**)
- Bag of words with combination of TFIDF and opinion weights. We took only the first thousand of the most probable opinion words (**BoW+tfidf+opweigh1000**).
- Bag of words with combination of TFIDF and opinion word weights. We took only first ten thousand of the most probable opinion words (**BoW+tfidf+opweight10000**).
- Bag of words with combination of TFIDF and opinion word weights. We took opinion weights for all words from the bag of words (**BoW+tfidf+opweight**).

For all the other pairs of tasks and domains we submitted only one run with **BoW+tfidf+opweight** set of features.

Besides we continued our study of the proposed tasks after the ROMIP deadlines and present our unofficial runs (in italic) in the same tables.

To obtain our first unofficial run 1,393 review duplicates were excluded from our book review collection. On the basis of such collection we obtained slightly better results. We marked such runs with *"nodupl"* postfix in the result tables.

Further we were interested to compare the results of our algorithm trained on the ROMIP data collections with the results of the algorithm trained on our data collections. In this way we retrained the classification model in each domain and evaluated it. These results were marked with *"romip"* postfix in corresponding tables.

4.1. Official metrics

There were a large amount of available metrics for evaluation [Chetviorkin et al., 2012]. To evaluate the performance of our algorithm we used *macro_precision, macro_recall, macro_F-measure, accuracy and average Euclidian distance*.

In addition two evaluation schemes were offered:

- **AND**, in evaluation involved only those reviews, which had the same score from both assessors (only for two-class classification)
- **OR,** we considered the answer of the algorithm as the right one if it matched with at least one of the assessors.

4.2. Three-class task

We started our study of sentiment classification with the three-class classification task. We had the best results in the classification of reviews about digital cameras and movies accordingly to accuracy and macro_F measures. In the book domain our algorithm was the second one accordingly to macro_F and fifth accordingly to the accuracy. The results can be found in Table 3. Our submissions are underlined; the best official results are in bold.

Four out of five of our runs in the movie domain had no statistically significant differences (Wilcoxon signed-rank test/Two-tailed test, $\alpha = 0.05$), and the result of one of them was considerably worse. Thus TFIDF word weights were very important for the quality of the classification but the amount of opinion words had no crucial meaning.

The exclusion of book review duplicates had improved all primary measures. In this case our macro_F result was the best in the book domain. Training on ROMIP collections gave roughly the same results in book and camera domains, but worse results in the movie domain. We discuss these differences in Section 4.5.

Table 1. Three-class classification results (OR)

Run_ID	*Object*	*Macro_Prec*	*Macro_Rec*	*Macro_F*	*Accuracy*
xxx-3	book	0.677	0.532	**0.577**	0.756
xxx-43 tfidf_op	book	0.671	0.517	0.570	0.756
xxx-11	book	0.658	0.475	0.488	**0.771**
Baseline	book	0.227	0.333	0.270	0.68
tfidf_op nodupl	*book*	*0.679*	*0.525*	*0.578*	*0.76*
tfidf_op romip	*book*	*0.664*	*0.510*	*0.571*	*0.76*
yyy-3 tfidf_op	camera	0.843	0.594	**0.663**	**0.841**
yyy-11	camera	0.797	0.596	0.661	0.815
Baseline	camera	0.216	0.333	0.262	0.648

Run_ID	*Object*	*Macro_Prec*	*Macro_Rec*	*Macro_F*	*Accuracy*
tfidf_op romip	*camera*	*0.804*	*0.598*	*0.658*	*0.837*
zzz-10 tfidf_op	film	0.671	0.535	**0.592**	**0.754**
zzz-19 tfidf_op1000	film	0.657	0.526	0.583	0.754
zzz-9 tfidf_op10000	film	0.660	0.524	0.582	0.751
zzz-1 tfidf	film	0.661	0.524	0.584	0.751
zzz-18 op_weight	film	0.585	0.431	0.494	0.635
Baseline	film	0.235	0.333	0.276	0.705
tfidf_op romip	*film*	*0.582*	*0.425*	*0.487*	*0.629*

4.3. Two-class task

In this task our results were the second by two primary measures in the camera domain (and first after training on the ROMIP collection) and second by macro_F in the movie domain (after training on the ROMIP collection we have lower results, see Section 4.5). In the book domain the results were rather low, but after training on the ROMIP collection the best macro_F result was obtained. The removal of duplicate reviews from the book collection had no effect in this task.

Table 4 shows our results and best two runs for each entity for evaluation schema OR in terms of macro f-measure and accuracy, our runs are underlined and unofficial runs are in italic.

Table 2. Two-class classification results (OR)

Run_ID	*Object*	*Macro_Prec*	*Macro_Rec*	*Macro_F*	*Accuracy*
xxx-40	book	0.714	0.804	**0.747**	0.895
xxx-0	book	0.751	0.721	0.735	0.924
xxx-24 (46)	book	0.968	0.630	0.690	**0.938**
xxx-19	book	0.790	0.651	0.694	0.931
xxx-35 tfidf_op	book	0.682	0.851	0.720	0.851
Baseline	book	0.46	0.5	0.479	0.92
tfidf_op nodupl	*book*	*0.682*	*0.851*	*0.720*	*0.851*

Run_ID	*Object*	*Macro_Prec*	*Macro_Rec*	*Macro_F*	*Accuracy*
tfidf_op romip	*book*	*0.710*	*0.852*	*0.751*	*0.876*
yyy-24	camera	0.918	0.940	**0.929**	**0.959**
yyy-16 tfidf_op	camera	0.944	0.898	0.919	0.956
Baseline	camera	0.426	0.5	0.46	0.852
tfidf_op romip	*camera*	*0.931*	*0.945*	*0.938*	*0.963*
zzz-23	film	0.776	0.797	**0.786**	**0.881**
zzz-9 tfidf_op	film	0.706	0.794	0.730	0.812
zzz-14	film	0.743	0.597	0.623	0.860
Baseline	film	0.427	0.5	0.461	0.854
tfidf_op romip	*film*	*0.682*	*0.790*	*0.685*	*0.742*

4.4. Five-class task

The five class evaluation scheme is very widespread in the Internet (five stars system), but a five-class sentiment classification is a rather difficult problem because we need not only to determine a text sentiment, but also to show its strength (the rating-inference problem).

Primary measures here were the accuracy and the average Euclidian distance. We achieved the best result accordingly to the accuracy measure in the movie domain and the second result in the book domain. After training on the book collection without duplicate reviews our algorithm gained the best accuracy result. On the ROMIP book collection the quality dropped significantly (see Section 4.5).

In the digital camera domain our results were quite low. Partly it could be explained by utilization of pros and cons by the other participants and differences in training collections. In our collection there was no strictly negative class (see Section 2).

Table 3. Five-class classification results (OR)

Run_ID	*Object*	*Avg_Eucl_Distance*	*Macro_F*	*Accuracy*
xxx-7	book	**0.872**	0.284	**0.622**
xxx-4 (9)	book	0.892	**0.291**	0.622
xxx-5 tfidf_op	book	0.972	0.270	0.615
Baseline	book	0.909	0.123	0.48

Run_ID	*Object*	*Avg_Eucl_Distance*	*Macro_F*	*Accuracy*
tfidf_op nodupl	*book*	*0.953*	*0.281*	*0.629*
tfidf_op romip	*book*	*1.04*	*0.201*	*0.542*
yyy-1	camera	**0.928**	0.298	0.567
yyy-3	camera	0.940	0.287	0.570
yyy-4	camera	0.971	**0.342**	**0.626**
yyy-2	camera	1.215	0.332	0.626
yyy-9 tfidf_op	camera	1.203	0.193	0.485
Baseline	camera	1.165	0.144	0.563
tfidf_op romip	*camera*	*1.125*	*0.234*	*0.530*
zzz-1 (5)	film	**1.026**	**0.286**	0.599
zzz-1	film	1.071	0.266	0.559
zzz-6 tfidf_op	film	1.133	0.247	**0.602**
Baseline	film	1.460	0.135	0.506
tfidf_op romip	*film*	*1.107*	*0.268*	*0.593*

4.5. The differences between collections

To substantiate the differences between the results obtained by our algorithm trained on different collections in one domain we decided to conduct some additional statistical research.

In the digital camera domain performance of the algorithm trained on our collection was worse than on ROMIP collection. We connect this gap with the differences in the review score distributions. (class "1" frequency, Section 2).

For the book and movie domains we had calculated the share of reviews in each class accordingly to the mapping scheme for a two-class task (for three class and five-class tasks results are the similar) and compared it with assessors' score distribution (OR evaluation scheme). We underlined the distribution that was more similar to the assessors.

Table 4–5. Score distribution in the train collections

Movie	**1**	**2**
Our	0.36	0.64
ROMIP	0.43	0.57
Eval	0.19	0.81

Book	**1**	**2**
Our	0.33	0.67
ROMIP	0.29	0.71
Eval	0.11	0.89

Thus the score distribution similarity between the train and test collections is highly correlated with the quality of review classification. The size of train collection has low influence on the quality of classification if the score distributions differ significantly.

5. Conclusions

This work is based on our previous research about influence of various features on the three-way review classification quality. In this study we describe the contribution of word weights to the quality of the three-class movie review classification. Then we apply the algorithm with the complete set of features to the other domains and tasks. Our approach demonstrates the good quality of classification in almost all domain-task pairs.

In addition we studied the dependence of the classification quality on the training collection. The similarity of the train and test collection score distributions played here a key role.

Acknowledgements. This work is partially supported by RFBR grant N11-07-00588-a.

References

1. *Ageev M., Dobrov B., Loukachevitch N., Sidorov A.* Experimental algorithms vs. basic line for web ad hoc, legal ad hoc, and legal categorization in RIRES2004 (in Russian). Proceedings of the Russian Information Retrieval Evaluation Seminar. Saint-Petersburg, 2004, pp. 62–89.
2. *Chetviorkin I., Braslavskiy P., Loukachevitch N.,* 2012 Sentiment Analysis Track at ROMIP 2011 (In this volume). Komp'iuternaia Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoi Konferentsii "Dialog 2012" [Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012"]. Bekasovo, 2012.
3. *Chetviorkin I. and Loukachevitch N.* Three-way movie review classification. Komp'iuternaia Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoi Konferentsii "Dialog 2011" [Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2011"]. Bekasovo, 2011a, 168–177.
4. *Chetviorkin I. and Loukachevitch N.* Extraction of Domain-specific Opinion Words for Similar Domains. Proceedings of the Workshop on Information Extraction and Knowledge Acquisition (IEKA 2011). Hissar, Bulgaria. 2011b. 7–12.
5. *Goldberg A., Zhu X.* Seeing stars when there aren't many stars: Graphbased semi-supervised learning for sentiment categorization. HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing. New York, 2006, pp. 45–52.

6. *Loukachevitch N. V., Chetviorkin I. I.* (2011) Extraction and use of opinion words for the three-way review classification problem (in Russian). Numerical Methods and Programming, Vol. 12, pp. 73–81
7. *Pang B., Lee L.* (2008) Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval. Hanover, Massachusetts, Now Publishers.
8. *Pang B., Lee L.* Seeing stars: Exploiting class relationships for sentiment categorization with respect of rating scales. Proceedings of the ACL, 2005. pp. 115–124.
9. *Pang, B., Lee, L., and Vaithyanathan, S.,* Thumbs up? Sentiment Classification Using Machine Learning Techniques. In Proceedings of EMNLP, 2002,
10. *Fan R.-E. , Chang K.-W., Hsieh C.-J., Wang X.-R., and Lin C.-J.* (2008), LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research, Vol. 9. pp. 1871–1874. Software available at http://www.csie.ntu.edu.tw/∼cjlin/liblinear
11. *Whitelaw C., Garg N., Argamon S.*: Using Appraisal Taxonomies for Sentiment Analysis. In: Proceedings of CIKM, Bremen, 2005.

АВТОМАТИЧЕСКИЙ АНАЛИЗ ТОНАЛЬНОСТИ ТЕКСТОВ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Котельников Е. В. (kotelnikov.ev@gmail.com),
Клековкина М. В. (klekovkina.mv@gmail.com)

Вятский государственный гуманитарный университет,
Киров, Россия

В статье представлены методы автоматической обработки текстов и машиного обучения, использованные авторами для решения задачи анализа мнений в рамках семинара РОМИП-2011. Обсуждаются вопросы выбора оптимального варианта векторной модели представления текстов и наиболее подходящего метода машинного обучения. Рассматриваются варианты построения векторной модели на основе подхода TF.IDF без использования обучающей информации о принадлежности текста тому или иному классу (unsupervised TF.IDF) и с использованием этой информации (supervised TF.IDF). Приведены данные о результатах применения следующих методов машинного обучения: наивного байесовского классификатора, метода Rocchio, метода k ближайших соседей, машин опорных векторов (SVM), метода на основе ключевых слов и его комбинации с SVM. Эксперименты показали, что наилучшие результаты показывает бинарная модель с косинусной нормализацией без обучения и метод, комбинирующий использование ключевых слов и SVM. Результаты экспериментов приводятся и анализируются в статье в сравнении с результатами другими участниками РОМИП-2011.

Ключевые слова: анализ тональности, машинное обучение, метод опорных векторов, метод Байеса

1. Введение

Автоматическая классификация текстов по тональности (анализ мнений, sentiment analysis) становится все более важной задачей, как с теоретической, так и с прикладной точек зрения [11]. На семинаре РОМИП-2011 впервые были предложены дорожки анализа отзывов пользователей по трем группам товаров — цифровые фотокамеры, книги и фильмы. Требовалось построить классификаторы для трех шкал оценок: двухбалльной, трехбалльной и пятибалльной.

Целью нашего участия в РОМИП-2011 являлось тестирование и сравнение, во-первых, различных подходов к представлению текста в рамках векторной модели, во-вторых, нескольких методов машинного обучения, в том числе метода опорных векторов (Support vector machine, SVM), наивного байесовского классификатора, метода классификации на основе ключевых слов и его комбинации с SVM.

В начале исследования мы ставили перед собой следующие вопросы:

1. Какой вариант векторной модели лучше подходит для решения задачи анализа мнений?
2. Какой метод машинного обучения лучше подходит для решения задачи анализа мнений?
3. Каким образом влияет размер оценочной шкалы (количество классов) на качество классификации?
4. Влияет ли тематика отзывов на качество классификации?

Для оценки качества классификации в процессе исследования использовались различные наборы данных. До того момента, когда тестовые данные, размеченные экспертами РОМИП, стали доступны, мы применяли скользящий контроль (cross-validation) на обучающих данных, предоставленных организаторами, и использовали подмножество тестовых данных, размеченных нами самостоятельно (по 100 отзывов по каждой группе товаров). После получения отзывов с экспертными оценками мы проверяли на них предварительные результаты — степень совпадения оказалась очень высокой.

Статья состоит из следующих разделов: в разделе 2 приводятся сведения о предварительной обработке текстов, в разделе 3 обсуждаются итоги исследования различных способов построения векторной модели текста. Раздел 4 посвящен используемым методам машинного обучения. В разделе 5 результаты экспериментов анализируются и сравниваются с результатами других участников. В разделе 6 обсуждаются выводы, сделанные на основе проведенных исследований, и направления дальнейшей работы.

2. Предварительная обработка

В наших исследованиях все используемые тексты подвергались единообразной предобработке. Из каждого текста исключались англоязычные и русскоязычные «стоп-слова» (частицы, предлоги, местоимения), удалялись слова длиной менее трех символов. Все слова преобразовывались к словарной форме (лемме) при помощи морфологического анализатора *mystem* от компании Яндекс. При этом из рассмотрения исключались все леммы, которые встречались менее чем в трех документах.

Полученная совокупность лемм обучающей коллекции составляет множество признаков для методов классификации и формирует словарь коллекции. Кроме лемм, в качестве признаков в словарь были добавлены различные варианты положительных и отрицательных смайликов — графических символов эмоционального отношения.

3. Векторная модель текста

Для ответа на первый вопрос («какой вариант векторной модели лучше подходит для решения задачи анализа мнений?») использовались два подхода

к построению векторной модели — без использования обучающей информации о принадлежности текста тому или иному классу (*unsupervised*) и с использованием этой информации (*supervised*) [2].

В обоих подходах вес слова в тексте определяется по схеме *TF.IDF* [13]:

$$t_{ik} = L_{ik} \cdot G_i \cdot D_k \qquad (1)$$

где t_{ik} — вес i-го термина в k-м документе,

L_{ik} — локальный вес i-го термина в k-м документе, отражающий значимость термина для данного документа,

G_i — глобальный вес i-го термина, отражающий значимость термина для всей коллекции,

D_k — нормализация для k-го документа.

Выражение (1) задает общую схему взвешивания, при подстановке в которую формул для всех трех компонентов получаются конкретные схемы вычисления весов. Для unsupervised TF.IDF мы исследовали следующие варианты [1]:

1) для локального веса: бинарный (BNRY), частотный (FREQ), логарифм частоты (LOGA);
2) для глобального веса: константный единичный (ONE), инвертированная документная частота (IDF), глобальный частотный IDF (GFIDF), логарифм GFIDF (IGFL). Кроме того, исследовался вариант вычисления глобального веса по методу TextRank [10];
3) для нормализации: отсутствие нормализации (NONE), косинусная нормализация (COSN).

Всего для unsupervised TF.IDF было протестировано $3 \times 5 \times 2 = 30$ способов вычисления весов терминов и получены следующие результаты (на основе метрики *macro F1* для бинарной классификации методом опорных векторов):

1) для разных групп товаров лучшими оказались разные способы вычисления локального веса: для фотокамер — FREQ, для фильмов — BNRY, для книг — LOGA и BNRY, причем для фотокамер отличие BNRY от FREQ не превышало 1 %;
2) во всех случаях лучшие результаты показал метод вычисления глобального веса ONE (присвоение всем терминам единичного глобального веса);
3) во всех случаях оказалось эффективнее вычислять косинусную нормализацию, чем обходиться без неё.

Для подхода supervised TF.IDF был выбран метод TF.RF, показавший по данным [6] наилучшие результаты в задаче тематической классификации. При этом в качестве локальных весов использовались методы взвешивания BNRY, FREQ и LOGA, осуществлялась косинусная нормализация, а глобальный вес подсчитывался по методу RF, предложенном в [5].

В методе RF (Relevance Frequency — релевантная частота) для вычисления глобального веса термина используется информация о распределении этого

термина по документам обучающей коллекции с учетом принадлежности документов к классам.

Обозначим a — количество документов, содержащих i-й термин и относящихся к классу C, b — количество документов, содержащих термин и не относящихся к классу C. Тогда, значимость i-го термина для класса C будет выражаться формулой [6]:

$$RF_i^C = \log_2\left(2 + \frac{a}{\max(1, b)}\right) \quad (2)$$

Результаты экспериментов показали, что метод вычисления глобальных весов RF показывает сходную эффективность с методом ONE — лучшим для unsupervised TF.IDF, — иногда незначительно превосходя его. Однако вычислительная сложность метода RF (как и всех других supervised методов) делает его применение нецелесообразным.

Таким образом, ответом на наш первый вопрос будТаким образом, ответом на наш первый вопрос будет утверждение, что с точки зрения эффективности и вычислительной сложности в качестве схемы взвешивания выгоднее всего использовать схему BNRY×ONE×COSN, т. е. бинарную модель с косинусной нормализацией. Такой вывод согласуется с результатами, полученными в [12].

4. Методы классификации

Для классификации текстов использовались известные методы машинного обучения [14]: наивный байесовский классификатор [7], метод Rocchio [3], метод k ближайших соседей [9], метод опорных векторов [4]. Кроме того, тестировался метод на основе ключевых слов и его комбинация с SVM.

В ходе предварительного тестирования на основе скользящего контроля по обучающим данным и размеченных самостоятельно тестовых документов выяснилось, что *методы Rocchio* и *k ближайших соседей* показывают существенно худшие характеристики качества, чем остальные. Поэтому было решено не отправлять на централизованное тестирование результаты, полученные этими методами.

Наивный байесовский классификатор был реализован традиционным образом [7], с учетом предварительной обработки текстов (см. раздел 2).

В качестве реализации *метода опорных векторов* была выбрана библиотека LIBSVM [8]. Проводился выбор ядра и подбор оптимальных параметров. Наилучшие результаты показало линейное ядро с регулирующим параметром $C = 1$.

Для задач с тремя и пятью классами использовалась стратегия «один против всех», когда обучается N классификаторов, где N — количество классов. Если несколько классификаторов «узнавали» тестовый документ, для окончательного решения выбирался наиболее положительный класс (при этом учитывалось неравномерное распределение количества обучающих отзывов по классам со смещением в сторону положительных оценок).

В *методе на основе ключевых слов* применялся лексико-статистический анализ и для каждого класса составлялся свой список ключевых слов. С этой целью для каждого слова из словаря коллекции (составленного после предварительной обработки, рассмотренной в разделе 2) вычислялся вес для каждого класса по методу RF (2). В список заносилось подмножество слов с наибольшим весом, пороговый вес определялся экспериментально, на основе скользящего контроля и метрики *macro F1*, отдельно для каждого класса.

Определение класса документа из тестовой коллекции осуществлялось следующим образом. Для каждого класса на основе его списка ключевых слов подсчитывается суммарный вес входящих в документ слов, таким образом, получался вес класса. Решение об отнесении документа к тому или иному классу принималось на основе сравнения весов классов.

Подобная идея реализована, например, в [12], но без вычисления весов и порогов отбора слов; также слова отбирались в список на основе простой частоты встречаемости в документах соответствующего класса.

В *методе, комбинирующем SVM и метод ключевых слов,* сначала независимо вычислялись гипотезы обоих методов об отнесении тестового документа к тому или иному классу. Итоговое решение в различных ситуациях вырабатывалось на основе следующей стратегии:

1) *ни один из методов не определил класс* — относили отзыв к наиболее положительному классу в данной задаче;
2) *класс определен только в одном из методов* — относили отзыв к данному классу;
3) *оба метода определили классы* — здесь возможны следующие варианты:
   - есть совпадение ответов одного из классификаторов SVM с ответами метода ключевых слов — относили отзыв к этому классу;
   - вес класса в методе ключевых слов превышал заданный порог (определенный эмпирически) — относили отзыв к этому классу;
   - ни одно из предыдущих условий не выполнялось — приписывали отзыву наиболее положительную оценку SVM.

5. Результаты экспериментов

Результаты тестирования методов для бинарной классификации представлены на рис. 1–3. Приведены метрики *macro F1* и *Accuracy* наших методов и нескольких лучших участников при схеме оценки AND. В большинстве случаев оценки по схеме OR не изменяют относительного расположения результатов.

Обозначения рассмотренных нами методов: *SVM* — метод опорных векторов, *KW (Keywords)* — метод ключевых слов, *Comb* — комбинированный метод, *NB (Naïve Bayes)* — наивный байесовский классификатор;

yyy-N — коды наших результатов, *xxx-N* — коды результатов других (лучших) участников.

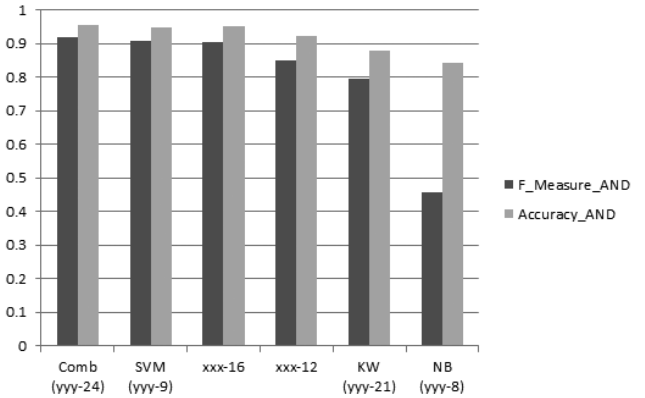


Рис. 1. Результаты классификации группы товаров «Фотокамеры» (AND)

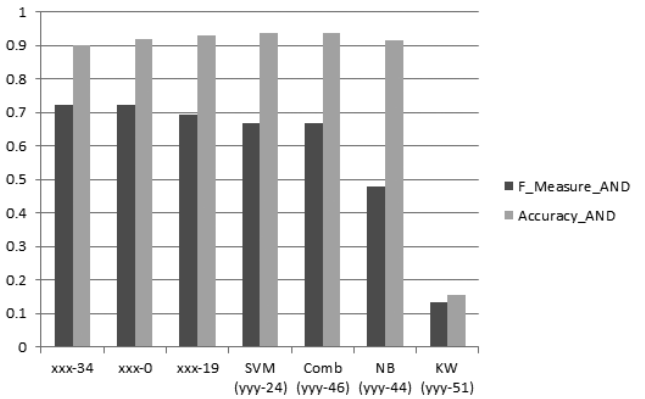


Рис. 2. Результаты классификации группы товаров «Книги» (AND)

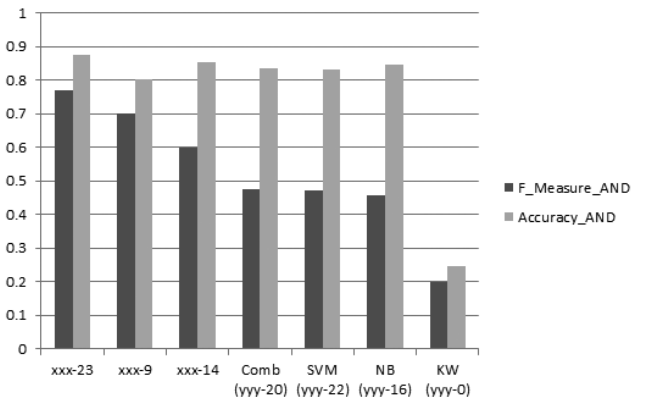


Рис. 3. Результаты классификации группы товаров «Фильмы» (AND)

Для задачи классификации с тремя и пятью классами результаты представлены в табл. 1 и 2. Приведены метрики *macro Precision*, *macro Recall macro F1* и *Accuracy* по схеме AND, обозначения аналогичны используемым на рисунках.

Таблица 1. Результаты классификации для трехбалльной шкалы (AND)

Run_ID	*Object*	*Macro_Prec*	*Macro_Rec*	*Macro_F1*	*Accuracy*
xxx-11	camera	0.745	**0.550**	**0.614**	0.787
xxx-3	camera	0.791	0.545	0.603	**0.812**
KW (yyy-12)	camera	0.753	0.514	0.574	0.778
Comb (yyy-6)	camera	**0.822**	0.515	0.566	0.797
SVM (yyy-1)	camera	0.590	0.377	0.412	0.720
xxx-43	book	**0.650**	**0.493**	**0.550**	**0.754**
xxx-3	book	0.641	0.492	0.536	0.715
Comb (yyy-37)	book	0.354	0.341	0.316	0.667
KW (yyy-47)	book	0.319	0.325	0.225	0.351
SVM (yyy-44)	book	0.232	0.293	0.259	0.636
xxx-10	film	**0.604**	**0.474**	**0.530**	**0.734**
xxx-19	film	0.598	0.471	0.527	**0.734**
Comb (yyy-4)	film	0.295	0.326	0.285	0.681
SVM (yyy-5)	film	0.233	0.309	0.265	0.662
KW (yyy-13)	film	0.300	0.285	0.206	0.312

Таблица 2. Результаты классификации для пятибалльной шкалы (AND)

Run_ID	*Object*	*Macro_Prec*	*Macro_Rec*	*Macro_F1*	*Accuracy*
xxx-4	camera	**0.591**	**0.223**	**0.259**	**0.520**
xxx-7	camera	0.393	0.195	0.246	0.493
Comb (yyy-3)	camera	0.582	0.206	0.225	0.473
KW (yyy-1)	camera	0.546	0.192	0.223	0.459
SVM (yyy-5)	camera	0.237	0.102	0.103	0.311
xxx-7	book	**0.510**	**0.225**	**0.253**	**0.574**
xxx-4	book	0.468	0.219	0.247	0.564
Comb (yyy-8)	book	0.285	0.184	0.204	0.468
SVM (yyy-6)	book	0.156	0.070	0.097	0.319
KW (yyy-2)	book	0.194	0.134	0.090	0.229
xxx-1	film	**0.325**	**0.194**	**0.230**	**0.531**
xxx-5	film	**0.325**	**0.194**	**0.230**	**0.531**
Comb (yyy-8)	film	0.201	0.095	0.110	0.258
KW (yyy-7)	film	0.197	0.091	0.081	0.191
SVM (yyy-4)	film	0.171	0.027	0.044	0.113

Проанализируем результаты классификации для двухбалльной шкалы (см. рис. 1–3), а также для трехбалльной и пятибалльной шкал (см. табл. 1, 2).

1. Из приведенных диаграмм видно, что метод опорных векторов показывает высокие значения метрики *F1* (за исключением группы товаров «Фильмы») и *Accuracy* (лучший результат для группы товаров «Книги»).

Для количества классов больше двух результаты метода опорных векторов существенно снижаются и он оказывается примерно в середине таблицы участников.

2. Наивный байесовский классификатор во всех случаях двухклассовой классификации показал низкие результаты по *F1*, но сопоставимые с лучшими результаты по *Accuracy*.

По техническим причинам в многоклассовой классификации наивный байесовский классификатор не был задействован.

3. Метод ключевых слов в бинарной классификации почти всегда показывает плохие результаты по обеим метрикам (за исключением группы товаров «Фотокамеры»).

В многоклассовых задачах ситуация неоднозначная, иногда метод ненамного отстает от лидеров и имеет преимущество перед SVM, в других случаях оказывается внизу таблицы результатов.

4. Результаты комбинированного метода для бинарной классификации практически идентичны методу опорных векторов, но в некоторых случаях (группа товаров «Фотокамеры») помогает скомпенсировать ошибки SVM и за счет этого выходит на первое место.

В случае трехбалльной и пятибалльной шкал комбинированный метод всегда показывает существенно лучшие результаты, чем метод опорных векторов и метод ключевых слов, и для группы товаров «Фотокамеры» имеет незначительную разницу по сравнению с лидерами.

В целом можно сделать следующие выводы.

1. SVM и комбинированный метод имеют, как правило, высокую точность (Precision), но низкую полноту (Recall), что в целом дает не слишком хорошую метрику F1. В свою очередь, например, для бинарной классификации низкая полнота получается из-за плохого распознавания отрицательных примеров. Связано это, возможно, с гораздо меньшим объемом обучающей выборки для негативных отзывов.

2. При увеличении количества классов результаты классификации всех участников семинара серьезно ухудшаются (например, для фотокамер при переходе от двух классов к пяти лучший результат по *F1* снижается с 92% до 26%). С другой стороны и оценки экспертов оказываются гораздо сильнее несогласованными в случае количества классов больше двух. В [11, стр. 27] высказывается мнение, что в отличие от многоклассовой тематической классификации в задаче анализа тональности текста, возможно, следует использовать регрессионные методы.

3. Результаты классификации отзывов для различных видов товаров довольно сильно отличаются. В табл. 3 приведены максимальные и средние значения по всем участникам метрик Precision, Recall, F1 и Accuracy. Из таблицы видно, что классификация отзывов по фотокамерам оказалась существенно проще. Возможно, это отчасти связано с тем, что в отзывах по данному виду товаров отдельно выделяются преимущества и недостатки товара, что более четко его характеризует. Другие причины обсуждаются, например в [11, стр. 37].

Таблица 3. Максимальные и средние значения Precision, Recall, F1 и Accuracy для бинарной классификации (AND)

	Precision		**Recall**		**F1**		**Accuracy**	
Группа товаров	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**
Фотокамеры	0,990	0,747	0,934	0,769	0,921	0,722	0,957	0,815
Книги	0,687	0,560	0,763	0,600	0,723	0,589	0,936	0,792
Фильмы	0,760	0,595	0,781	0,614	0,769	0,545	0,875	0,674

6. Заключение

Проведенное исследование позволило нам ответить на заданные в начале вопросы.

1. «Какой вариант векторной модели лучше подходит для решения задачи анализа мнений?» — бинарная модель с косинусной нормализацией без глобальных весов.
2. «Какой метод машинного обучения лучше подходит для решения задачи анализа мнений?» — среди исследованных нами методов наилучшие результаты показал метод, комбинирующий методы опорных векторов и ключевых слов.
3. «Каким образом влияет размер оценочной шкалы (количество классов) на качество классификации?» — при увеличении диапазона шкалы качество классификации существенно ухудшается.
4. «Влияет ли тематика отзывов на качество классификации?» — качество классификации в большой степени зависит от тематики отзывов.

В целом, наш первый опыт участия в семинаре РОМИП следует признать успешным: на предоставленных организаторами тестовых материалах удалось провести задуманное исследование, при централизованной оценке наши результаты по нескольким прогонам оказались на первом месте.

В дальнейшем предполагается совершенствовать рассмотренные методы за счет использования специализированных словарей эмоциональной лексики и применения других методов машинного обучения — регрессионного и структурированного вариантов SVM, Gradient boosting.

Хочется надеяться, что на будущих семинарах РОМИП проблема анализа тональности текста останется в центре внимания и в её рамках будут предложены новые интересные задачи.

Литература

1. *Chisholm E., Kolda T. G.* New term weighting formulas for the vector space method in information retrieval. Technical Report Number ORNL-TM-13756, Oak Ridge National Laboratory, Oak Ridge, TN, March 1999.
2. *Debole F., Sebastiani F.* Supervised term weighting for automated text categorization. Proceedings of the 2003 ACM symposium on Applied computing SAC 03, 2003, Vol. 138(Ml), pp. 784–788.
3. *Joachims T.* A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. Proceedings of 14th International Conference on Machine Learning, Nashville, TN, 1997, pp. 143–151.
4. *Joachims T.* Text categorization with support vector machines: learning with many relevant features. Proceedings of 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 137–142.
5. *Lan M.* (2007) A New Term Weighting Method for Text Categorization. PhD Theses.
6. *Lan M., Tan C. L., Su J., Lu Y.* (2009), Supervised and Traditional Term Weighting Methods for Automatic Text Categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, no. 4, pp. 721–735.
7. *Lewis D. D.* Naive (Bayes) at forty: The independence assumption in information retrieval. Proceedings of 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 4–15.
8. *LIBSVM* — A Library for Support Vector Machines, available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
9. *Masand B., Linoff G., Waltz D.* Classifying news stories using memory-based reasoning. Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 1992, pp. 59–65.
10. *Mihalcea R., Tarau P. Textrank*: Bringing order into texts. Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004, pp. 404–411.
11. *Pang B., Lee L.* (2008), Opinion Mining and Sentiment Analysis, Foundations and Trends® in Information Retrieval, no. 2, pp. 1–135.
12. *Pang B., Lee L., Vaithyanathan S.* Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.
13. *Salton G., Buckley C.* (1988), Term-weighting approaches in automatic text retrieval, Information Processing & Management, Vol. 24, no. 5, pp. 513–523.
14. *Sebastiani F.* (2002), Machine learning in automated text categorization ACM Computing Surveys, Vol. 34, no. 1, pp. 1–47.

LANGUAGE INDEPENDENT APPROACH TO SENTIMENT ANALYSIS (LIMSI PARTICIPATION IN ROMIP'11)

Pak A. (alexpak@limsi.fr),
Paroubek P. (pap@limsi.fr)

Université Paris-Sud, Lab. LIMSI-CNRS, Bâtiment 508, F-91405 Orsay Cedex, France

Sentiment analysis is a challenging task for computational linguistics. It poses a difficult problem of identifying user opinion in a given text. In this paper, we describe participation of LIMSI in the sentiment analysis track of the Russian annual evaluation campaign (ROMIP'11). The goal of the track was classification of opinions expressed in blog posts into two, three, and five classes. Our system based on SVM with dependency graph and n-gram features was placed 1st in 5-class task on all three datasets (movies, books, cameras), 3rd in the 2-class task on the movies dataset, and 4th in the 3-class task on the cameras dataset, according to the official results.

Key words: sentiment analysis, polarity classification, SVM, dependency parsing

1. Introduction

Sentiment analysis is a recent field of computational linguistics which emerged due to the growing demand of analysis of social media and user generated content in the Internet. Hence, to encourage the research in this field and to discover the current state of the art, sentiment analysis tasks have been included in a set of traditional evaluation campaigns tracks in information retrieval (IR) and natural language processing (NLP). TREC[1] 2006 added a blog opinion mining track, SemEval[2] 2010 organized a task on polarity disambiguation of Chinese adjectives, I2B2[3] 2011 dedicated one of the tasks to sentiment classification in suicide notes. In this paper, we describe our participation in ROMIP[4] 2011 sentiment analysis track.

[1] Text Retrieval Conference: http://trec.nist.gov/

[2] Evaluation Exercises on Semantic Evaluation: http://semeval2.fbk.eu/

[3] Informatics for Integrating Biology and the Bedside: http://www.i2b2.org/NLP/

[4] Russian Information Retrieval Evaluation Seminar: http://romip.ru

1.1. Task description

ROMIP is an annual evaluation campaign in information retrieval launched in 2002 [3]. In ROMIP 2011, the organizers added the sentiment analysis track which aimed at classification of opinions in user generated content. A dataset composed ofproduct reviews collected from a recommendation service Imhonet[5] and product aggregator service Yandex.Market[6] was provided to participants for training their systems. The dataset contained reviews about three topics: digital cameras, books, and movies. Table 1 shows the characteristics of the dataset.

Table 1. Characteristics of the training dataset

Topic	Source	# of reviews
Books	Imhonet	24,159
Movies	Imhonet	15,718
Cameras	Yandex.Market	10,370

Each review consists of the text of the review and meta information. Meta information contains the rating score assigned to the product, the product ID, reviewer ID, and the review ID. Reviews from Yandex.Market also contain review creation time, usefulness ofthe review (assigned by other users), pros and cons of the product given by the review author. Inour work, we used only the review text, the score, and pros/cons if available. The score is given on 1–5 scale for Imhonet reviews, and 1–10 scale for Yandex.Market reviews, where a higher value represents more positive opinion. Figure 1 shows an example of a digital camera review.

The evaluation dataset was not provided until the evaluation phase at the end of the campaign. The organizers have collected 16 861 posts from LiveJournal[7] blogging platform that mention books, movies, or cameras out of which 874 posts were annotated by two human experts. What makes this track different from other evaluation campaigns, is that the evaluation dataset was not of the same nature as the training data. First, the texts had different genres (product reviews vs. blogposts), and secondly the annotations were produced differently: the training data was composed automatically, while the testdata was annotated manually. Figure 2 shows an example of a test document.

The track was divided into three subtracks:

- Opinion classification into two classes: negative/positive
- Opinion classification into three classes: negative/mixed/positive

[5] http://imhonet.ru

[6] http://market.yandex.ru

[7] http://livejournal.com

- Opinion classification into five classes: a score on the scale 1–5, where 1 represents an exclusively negative opinion, and 5 represents an exclusively positive opinion

In its turn, each subtrack had 3 runs by the number of topics: classification in each topic was evaluated separately, resulting in total 9 separate evaluations.

```
<row rowNumber="0">
  <value columnNumber="0">1328131</value>     <!-- review ID      -->
  <value columnNumber="1">926707</value>      <!-- product ID     -->
  <value columnNumber="2">48983640</value>    <!-- author ID      -->
  <value columnNumber="3">2009-05-03</value>  <!-- creation time  -->
  <value columnNumber="4">4</value>           <!-- rating         -->
  <value columnNumber="5">                    <!-- text           -->
    Хороший выбор для опытного фотолюбителя.
    <!-- A good choice for an experienced amateur photographer. -->
  </value>
  <value columnNumber="6">                    <!-- pros           -->
    Большой выбор режимов съемки,12-кратный оптический зум,
    естественная цветопередача,большой ЖК-экран.
    <!-- Large selection of shooting modes,12-times optical zoom,
    natural color, large LCD screen. -->
  </value>
  <value columnNumber="7">                    <!-- cons           -->
    Невысокая скорость подзарядки фотовспышки.
    <!-- The low speed of flash recharge. -->
  </value>
  <value columnNumber="8">0.59375</value>     <!-- usefulness     -->
</row>
```

Fig. 1. An example of a review from the training dataset. Russian text has been translated into English only for this example

1.2. Task challenge

Sentiment analysis is a difficult task even for resource-rich languages (read, English). Along with simple language processing, such as part-of-speech (POS) tagging, more sophisticated NLP tools such as discourse parsers and lexical resources may be required by existing approaches. Thus, it is quite difficult to adapt methods that were developed in other languages (read, English) to Russian.

The ROMIP track poses additional challenges other than the difficulty of analysing sentiments in general. As mentioned before, the evaluation set was not constructed the same way as the training data. That makes it more difficult for statistical based approaches as the language model differs in two datasets. Moreover, the distribution of classes is also different. The training set contained more positive reviews, however

the way the reviews were picked for annotation was unknown. Finally, the interpretation of rating also varies, as there were different conventions when assigning scoring products and when annotating the test set. In other words, a user of Yandex.Market may have a different interpretation of 3 stars assigned to a camera from a human annotator who rates a review.Multiclass classification was another challenge, since most of research on polarity classification consider it a binary problem, i. e. classifying a document into positive/negative classes.

```
<?xml version="1.0" encoding="windows-1251"?>
<document>
  <ID>11347</ID>
  <link>http://vikilt.livejournal.com/12619.html</link>
  <date>2011-02-06T20:59:15Z</date>
  <object>
    Плохая училка
    <!-- Bad teacher -->
  </object>
  <text>
    Недавно посмотрел фильм "Очень плохая училка" и наконец,
    увидел этого самого Джастина Тимберлейка о котором так много
    было звона и сильно удивился. В фильме персонаж Кэмерон Диос
    как только видит этого Джастина начинает млеть и интенсивно
    намокать, хотя сам персонаж никаких эротический эмоций кроме смеха
    и недоумения не вызывает. Дальше он там, в фильме поёт песенку,
    которая тоже оставляет желать лучшего. Девушки, неужели вам
    действительно нравятся такие чахлые додики сомнительной наружности?
    <!-- Recently, I have watched a movie "Bad teacher" and finally,
    I've seen this Justin Timberlake about whom there have been
    so much buzz and I was surprised a lot. In the movie,
    the character of Cameron Diaz becomes excited as soon as
    she sees this Justin, although his character does not invoke
    any feelings except laughing. Next, he there, in the movie,
    sings a song, which is poor also. Girls, do you really
    like such doubtful looking nerds? -->
  </text>
</document>
```

Fig. 2. An example of a document from the evaluation set. Russian text has been translated into English only for this example

Therefore, to tackle the problem, we have decided to use a language independent approach that is not dependent on sophisticated NLP tools or lexical resources (e. g. affective lexicons) that are not available in Russian. We used an SVM based system with features based on n-grams, part-of-speech tags, and dependency parsing. For that we have trained a dependency parser on the Russian National Corpus[8]. Additionally, a study on terms weighting and corpus composition has been performed in order

[8] http://www.ruscorpora.ru/en/

to optimize the performance of our system. The detailed description of our system is presented in Section 3 right after the overview of the current state of the art in Section 2. We report our experimental evaluation along with official results in Section 4. Finally, we draw conclusions in Section 5.

2. Related work

Polarity classification is one of the basic problems of sentiment analysis and probably the most studied. The existing approaches fall into two large categories: lexicon based and machine learning based methods.

Lexicon based methods make use of existing lexical resources that vary in their complexity starting from simple lists of positive and negative words to more sophisticated semantic maps. For English, one may use resources developed specifically for sentiment analysis and affective science such as SentiWordNet [4], WordNet-Affect [19], ANEW [2], General Inquirer [18], and also general purpose resources, such as WordNet [6]. However, to our knowledge no similar publicly available resource exists in Russian, therefore a lexicon based approach would require to create a lexicon from scratch which is a costly process. More over, the quality of the system would strongly depend on the quality of the developed resource. As the lexicon should cover well the analysed language model.

Machine learning based approaches in the majority are based on a classical framework for text classification. The most commonly used one is support vector machines with n-gram features trained on a large set of text with known polarities (usually positive or negative) [14][13].Other systems add on top of this basic framework additional text preprocessing, feature selection, and NLP.

The amount and the complexity of NLP varies in different approaches. We have previously reported the usefulness of POS tags for opinion mining [10]. Dependency parsing has been also widely used in the sentiment analysis domain for extracting additional features [1][8], determining opinion subject [21], and additional text analysis. A recent work by Zirn et al. [22], used discourse parsing to take into account relation between phrases for fine-grained polarity classification. One of few works on sentiment analysis in Russian by Pazelskaya and Solovyev [15] used a manually constructed affective lexicon along with POS-tagging and lexical parsing information for a rule based polarity classifier. However, many of these approaches are difficult to reproduce for the ROMIP track as there are few NLP tools for Russian that are publicly available.

3. Our approach

To overcome the difficulties of the task, thus to create a sentiment analysis system for Russian that would be robust in different topics without overfitting the training model, we developed an SVM based system using the LIBLINEAR package developed

by Fan et al. [5]. For the 2-class track we trained SVM in binary classification mode, for the 3 and 5-class tracks, we used a multiclass and regression modes.

3.1. Training dataset composition

The distribution of opinion scores in the training data set was highly unbalanced, which caused difficulties for training the model. Figure 3 shows distributions of reviews by scores in different topics. In general, positive reviews are prevailing in the training dataset which creates a bias towards a positive class. For the 2-class problem, we have decided to balance the training dataset by using an equal number of reviews of negative and positive opinions. Thus we considered books and movies reviews with scores 1–4 as negative and 9–10 as positive, and in the cameras collection,we considered reviews with scores 1–2 as negative and 5 as positive. The rest of the reviews were not included in the training. For 3-class and 5-class problems we left the dataset as is, because there would not be enough data to represent each class.

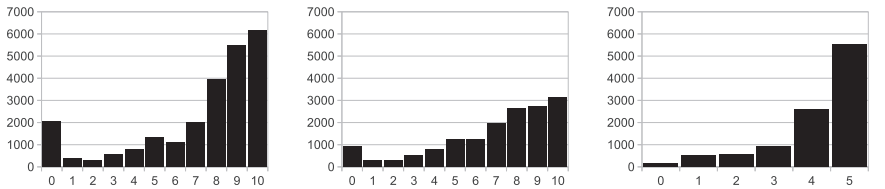


Fig. 3. Score distribution in books (left), movies (center), and cameras (right) datasets

Another decision which had to be made, was whether to train three separate modelsfor each topic orto combine all the data and to train one general model to classify reviews from each topic. We have experimented with both settings, and report the results in Section 4.

Reviews from Yandex.Market on cameras contain product prosand cons. To benefit from this additional information, we decided to include it in the text of there view. Thus, if a review is considered to be positive (using the criteria as mentioned above) then we add pros as the last phrase of the text. Otherwise, if a review is negative, we use cons. We have discovered that by doing this, we improved the accuracy of binary polarity classification up to 13.7%.

3.2. Feature vector construction

We have experimented with two types of features to build the model: traditional n-grams and our proposed d-grams features that are based on dependency tree of text sentences [12].

N-grams In the n-gram model, text is represented as a bag of words subsequences of a fixed size. We have experimented with unigrams and bigrams. Any non alphanumeric character was considered as a word boundary. Negations has been handled by attaching a negation particle (*не* — no, *ни* — neither, нет — not) to a preceding and a following word when constructing n-grams [10][20].

D-grams D-grams are similar to n-grams, however, while n-grams are constructed by splitting a text into subsequences of consecutive words, d-grams are constructed from a dependency parse tree, where words are linked by syntactic relations.

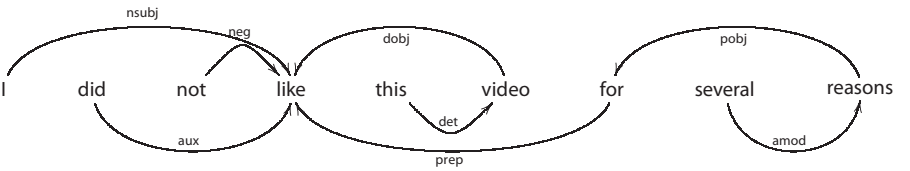


Fig. 4. Dependency graph of a sentence "The soundtrack was awful"

Figure 4 depicts an example of dependency parse tree of a sentence "The soundtrack was awful". The dependency relations that we obtain are as follows:

{(I, nsubj, like),
(did, aux, like),
(not, neg, like),
(this, det, video),
(video, dobj, like),
(for, prep, like),
(several, amod, reasons),
(reasons, pobj, for)}

They are served as features in our d-gram model replacing the traditional n-gram model.To obtain dependency parse trees, we first applied TreeTagger [16][17] for tokenization and POS-tagging. Next, we fed the tagged output to the MaltParser [9] that we had trained on the Russian National Corpora.

Weighting scheme We consider two weighting schemes which are used in sentiment analysis.

Binary weights were used in first experiments by Pang et al. [14] and proven to yield better results than traditional information retrieval weighting such as TF-IDF. It assigns equal importance to all the terms presented in a document:

$$w(g_i) = 1, \text{if} g_i \in d, \text{otherwise} = 0 \quad (1)$$

where g_i is aterm(n-gram), d is a document. **Delta TF-IDF** was proposed by Martineau et al. [7] and proven to be efficient by Paltoglou et al. [13], assigns more importance to terms that appear primarily in one set (positive or negative):

$$w(g_i) = \mathrm{tf}(g_i) \cdot \log \frac{\mathrm{df_p}(g_i) + 0.5}{\mathrm{df_n}(g_i) + 0.5} \tag{2}$$

where $\mathrm{tf}(g_i)$ is term-frequency of a term (number of times g_i appears in document D), $\mathrm{df_p}(g_i)$ is positive document frequency (number of times g_i appears in documents with positive polarity), $\mathrm{dfp}(g_i)$ is negative document frequency.

We augment Delta TF-IDF formula with our proposed average term-frequency normalization that lowers importance of words that are frequently used in a document [11]:

$$\mathrm{avg.tf}(g_i) = \frac{\sum_{\forall T, g_i \in T} \mathrm{tf(g_i)}}{\{T | g_i \in T\}} \tag{3}$$

where $\{T|g_i \in T\}$ is a set of documents containing term g_i. Thus, we modify Delta TF-IDF weight as follows:

$$w(g_i) = \frac{\mathrm{tf}(g_i)}{\mathrm{avg.tf}(g_i)} \cdot \log \frac{\mathrm{df_p}(g_i) + 0.5}{\mathrm{df_n}(g_i) + 0.5} \tag{4}$$

4. Experiments and results

In this section, we report results obtained during the system development phase and the offocial results provided by the organizers of ROMIP.All the development results were obtained after performing 10-fold cross validation.

4.1. Development results

Table 2. Macro-averaged accuracy over different training and test data. Rows correspond to a dataset on which the model has been trained, columns correspond to test data. ***Combined*** is a combination of all three topics

		Train data			
		books	movies	cameras	combined
Test data	books	76.0	74.0	65.5	73.4
	movies	77.3	76.4	66.4	74.5
	cameras	63.2	62.0	76.0	65.5
	combined	78.4	78.9	77.1	78.6

For the development phase, we present results only on binary classification as all the system parameters were tuned according to the results of these experiments.

Table 2 shows results of n-gram based model with binary weights across different topics. According top revious research on domain-adaptation for sentiment analysis a model trained on the same topics as the test set performs better than one trained on another topic. However, we were interested whether combining all the training data thus increasing the size of the available training data set improves the model. As we can see from the results, the model trained on the combined data performs better than a model trained only on one topic and the model trained on the same topic as the test set performs better than a model trained on another topic. However, we will see that it would change once we add additional information.

Table 3. Performance gain when adding class balancing and including pros/cons

	Books		**Movies**		**Cameras**	
	div	com	div	com	div	com
default	76.0	78.4	76.4	78.9	76.0	77.1
+ balanced	78.1 +1.9	79.5 +0.9	76.3 −0.1	78.2 −0.7	77.4 +1.4	77.5 +0.4
+ pros/cons	78.1	79.6 +0.1	76.3	78.6 +0.4	91.8 +13.7	87.9 +10.4

Table 3 shows show the performance changes after balancing the training data, and after adding pros and cons. Balancing the training set improves accuracy when classifying books and cameras and slightly degrades the performance on the movies collection. Adding pros and cons drastically improves the performance over the cameras test set (up to 13.7% of gain). Notice, also that the model trained only on the cameras collection performs much better than the one trained on combined data (91.8% vs. 87.9%). Thus, for the following experiments we keep these settings: balancing training set and including pros and cons.

Table 4. Classification accuracy across different topics. For each topic, we evaluated a model trained on the same topic (div) and a model trained on all the reviews (com)

	Books		**Movies**		**Cameras**	
	div	com	div	com	div	com
ngrams + binary	78.1	79.6	**76.3**	**78.6**	91.8	87.9
ngrams + Δtfidf	77.4	78.8	76.2	76.5	93.1	90.4
dgrams + binary	78.0	79.8	74.9	77.8	91.3	88.2
dgrams + Δtfidf	**78.4**	**80.2**	76.1	77.3	**93.6**	**91.3**

Table 4 shows the comparison of the model using different features and weighting schemes. Here we have compared the traditional n-grams model with our proposed d-grams features using the same weighting schemes (binary and Delta TF-IDF). As we observe from the results, d-grams with Delta TF-IDF yields better accuracy

on books and cameras test sets, while n-grams with binary weights perform better on the movies collection. However the difference is not very big.

4.2. Official results

According to the results we have obtained during the development phase, we have submitted the official runs on the unseen data. For 2-class track we have submitted 6 systems. For 3-class and 5-class tracks, we trained only systems based on n-grams due to time and resource constrains. For each of these tracks, we have submitted 4 systems. The summary of the submitted systems is presented in Table 6. The overall standings are depicted in Figures 5–7.

Table 5. Summary of the submitted systems

System ID	Mode	Features	Weights	Training set
2-class track				
2-class track	binary	d-grams	Δtfidf	divided
2-dgram-delta-com	binary	d-grams	Δtfidf	combined
2-ngram-delta-div	binary	n-grams	Δtfidf	divided
2-ngram-delta-com	binary	n-grams	Δtfidf	combined
2-ngram-bin-div	binary	n-grams	binary	divided
2-ngram-bin-com	binary	n-grams	binary	combined
3-class track				
3-ngram-bin-div	multiclass	n-grams	binary	divided
3-ngram-bin-com	multiclass	n-grams	binary	combined
3-regr-ngram-bin-div	regression	n-grams	binary	divided
3-regr-ngram-bin-com	regression	n-grams	binary	combined
5-class track				
5-ngram-bin-div	multiclass	n-grams	binary	divided
5-ngram-bin-com	multiclass	n-grams	binary	combined
5-regr-ngram-bin-div	regression	n-grams	binary	divided
5-regr-ngram-bin-com	regression	n-grams	binary	combined

5. Conclusions

Sentiment analysis is a challenging task for computational linguistics. It becomes especially difficult for resource-poor languages. In this paper, we have described our participation in Russian sentiment analysis evaluation campaign

Table 6. Official ranking of the submitted systems

System ID	Books		Movies		Cameras	
	score	rank	score	rank	score	rank
2-class track						
2-dgram-delta-div	65.1	24/53	70.3	5/27	**81.7**	**11/25**
2-dgram-delta-com	**66.1**	**23/53**	**70.9**	**3/27**	76.6	17/25
2-ngram-delta-div	61.8	31/53	70.0	7/27	77.8	15/25
2-ngram-delta-com	63.0	27/53	67.7	8/27	80.6	12/25
2-ngram-bin-div	57.9	36/53	63.7	10/27	79.2	13/25
2-ngram-bin-com	58.8	35/53	65.3	9/27	78.8	14/25
3-class track						
3-ngram-bin-div	**48.4**	**12/52**	47.7	9/21	55.7	8/15
3-ngram-bin-com	49.9	18/52	**50.4**	**5/21**	**62.6**	**4/15**
3-regr-ngram-bin-div	47.6	21/52	48.4	8/21	50.0	9/15
3-regr-ngram-bin-com	48.8	16/52	49.8	6/21	57.4	7/15
5-class track						
5-ngram-bin-div	27.0	4/10	24.6	5/10	**34.2**	**1/10**
5-ngram-bin-com	**29.1**	**1/10**	**28.6**	**1/10**	28.3	7/10
5-regr-ngram-bin-div	28.5	3/10	26.6	3/10	31.1	4/10
5-regr-ngram-bin-com	**29.1**	**1/10**	**28.6**	**1/10**	28.3	7/10

ROMIP 2011. We have tested our language independent framework for polarity classification that is based on SVM with the traditional n-grams model and our proposed features based on dependency parse trees. The developed system was ranked 1st in the 5-class track in all topics, 3rd in the 3-class track in movies domain, and 4th in the binary classification track in cameras domain according to the official evaluation metrics.

References

1. *S. Arora, E. Mayfield, C. Penstein-Ros´e, and E. Nyberg.* Sentiment classification using automatically extracted subgraph features. In Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, CAAGET '10, pages 131–139, Morristown, NJ, USA, 2010. Association for Computational Linguistics.
2. *M. M. Bradley and P. J. Lang.* Affective norms for English words (ANEW). Gainesville, FL. The NIMH Center for the Study of Emotion and Attention. University of Florida, 1999.
3. *B. Dobrov, I. Kuralenok, N. Loukachevitch, I. Nekrestyanov, and I. Segalovich*. Russian Information Retrieval Evaluation Seminar. In Proceedings of the Fourth International Conference on Language Resources and Evaluation,Lisbon,Portugal, May 2004.

4. *A. Esuli and F. Sebastiani.* SentiWordNet: A Publicly Available Lexical Resource forOpinionMining. InIn Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06), pages 417–422, 2006.
5. *R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin.* Liblinear: A library for large linear classification. J. Mach. Learn. Res., 9:1871–1874, June 2008.
6. *C. Fellbaum,* editor. WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, illustrated edition edition, May 1998.
7. *J. Martineau and T. Finin*. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In Proceedings of the Third AAAI Internatonal Conference on Weblogs and Social Media, San Jose, CA, May 2009. AAAI Press. (poster paper).
8. *T. Nakagawa, K. Inui, and S. Kurohashi.* Dependency tree-based sentiment classification using crfs with hidden variables. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10, pages 786–794, Morristown, NJ, USA, 2010. Association for Computational Linguistics.
9. *J. Nivre, J. Hall, and J. Nilsson*. MaltParser: A data-driven parser-generator for dependency parsing. In Proc. of LREC-2006, 2006.
10. *A. Pak and P. Paroubek*. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, may 2010. European Language Resources Association(ELRA).
11. *A. Pak and P. Paroubek*. Normalization of Term Weighting Scheme for Sentiment Analysis. In Proceedings of the 5th Language Technology Conference, Poznan, Poland, November 2011.
12. *A. Pak and P. Paroubek*. Text representation using dependency tree subgraphs for sentiment analysis. In Proceedings of the 16th international conference on Database systems for advanced applications, DASFAA'11, pages 323–332, Berlin,Heidelberg, 2011. Springer-Verlag.
13. *G. Paltoglou and M. Thelwall.* A study of information retrieval weighting schemes for sentimentanalysis. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pages 1386–1395, Morristown, NJ, USA, 2010. Association for Computational Linguistics.
14. *B. Pang, L. Lee, and S. Vaithyanathan*. Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing -Volume 10, EMNLP '02, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
15. *A. G. Pazelskaya and A. N. Solovyev.* A method of sentiment analysis in Russian texts. In Proceedings of the Dialog 2011 the 17th International Conference On Computational Linguistics, Moscow region, Russia, May 2011.
16. *H. Schmid.* Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proceedings of the International Conference on New Methods in Language Processing, pages 44–49, 1994.
17. *S. Sharoff, M. Kopotev, T. Erjavec, A. Feldman, and D. Divjak*. Designing and evaluating a russian tagset. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, may

2008 .European Language Resources Association(ELRA). http://www.lrecconf.org/proceedings/lrec2008/.
18. *P. J. Stone and E.B. Hunt*. A computer approach to content analysis: studies using the general inquirer system. In Proceedings of the May 21–23, 1963, spring joint computer conference, AFIPS'63 (Spring), pages 241–256, New York, NY, USA, 1963. ACM.
19. *C. V. Strapparava* and A. WordNet-Affect: an affective extension of WordNet. In Proceedings of the 4th International Conference on Language Resources and Evaluation , LREC, 2004.
20. *M. Wiegand, B. Roth, and D. Klakow*. A survey on the role of negation in sentiment analysis, 2010.
21. *L. Zhuang, F. Jing, and X.-Y. Zhu*. Movie review mining and summarization. In Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06, pages 43–50, New York, NY, USA, 2006. ACM.
22. *C. Zirn, M. Niepert, H. Stuckenschmidt, and M. Strube*. Fine-grained sentiment analysis with structural features. In Proceedings of 5th International Joint Conference on Natural Language Processing, pages 336–344, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

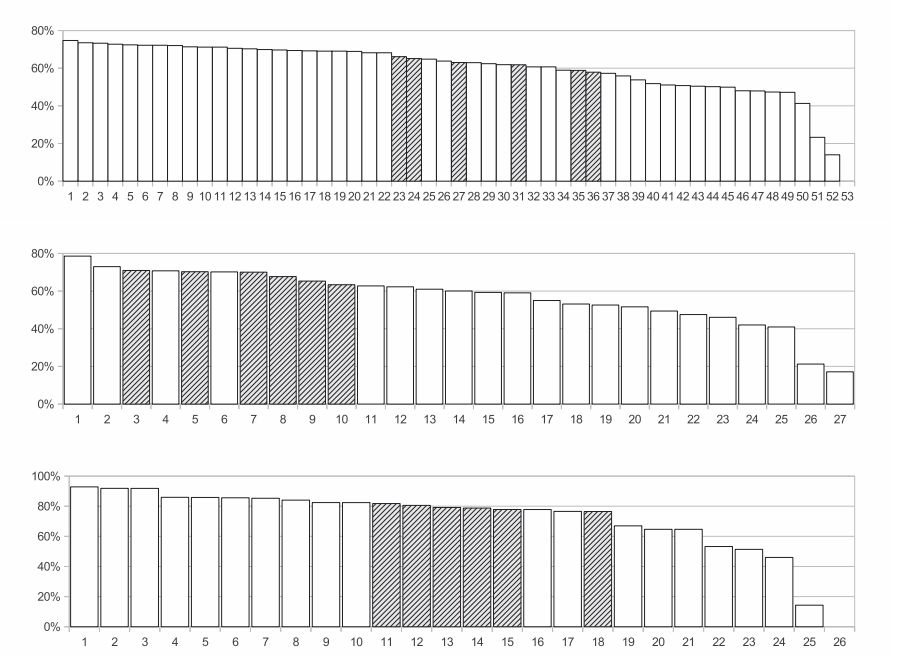


Fig. 5. Systems performance and ranking on the 2-class track on books (top), movies (middle), and cameras (bottom) collections. Our systems are highlighted

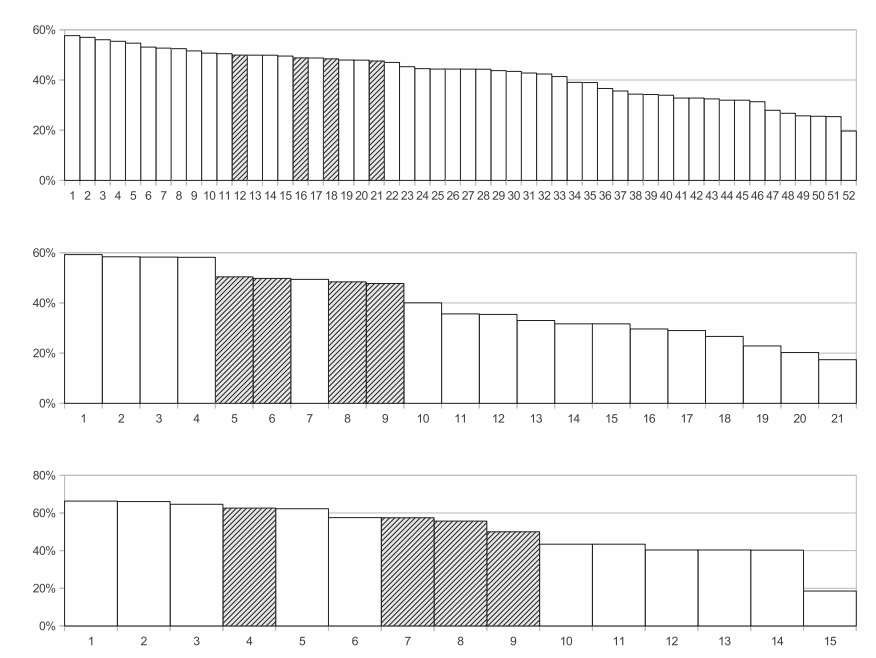


Fig. 6. Systems performance and ranking on the 3-class track on books (top), movies (middle), and cameras (bottom) collections. Our systems are highlighted

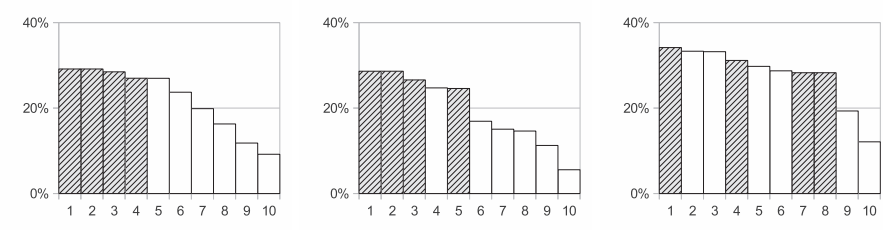


Fig. 7. Systems performance and ranking on the 5-class track on books (top), movies (middle), and cameras (bottom) collections. Our systems are highlighted

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ МЕТОДОВ ТЕМАТИЧЕСКОЙ КЛАССИФИКАЦИИ В ЗАДАЧЕ КЛАССИФИКАЦИИ ОТЗЫВОВ О КНИГАХ

Поляков П. Ю. (pavel@rco.ru),
Калинина М. В. (kalinina_m@rco.ru),
Плешко В. В. (volodia@rco.ru)

ООО «ЭР СИ О», Москва, Россия

В данной работе исследуются различные способы формирования обучающей выборки, методов извлечения классификационных признаков, а также методов построения классификаторов для решения задач классификации отзывов о книгах на 2 (положительный, отрицательный) и 3 (положительный, отрицательный и нейтральный) класса. Показано, что хороший результат можно получить путем применения к рассматриваемым задачам методов тематической классификации. Достигнутые показатели не уступают наилучшим результатам классификации Веб-сайтов и нормативно-правовых документов, полученным участниками семинара РОМИП.

Ключевые слова: анализ мнений, определение тональности, автоматическая классификация, машинное обучение, извлечение классификационных признаков, метод опорных векторов, регрессия

Введение

Задача автоматической классификации отзывов о товарах является на сегодняшний день весьма востребованной, о чем свидетельствует появление соответствующей функции в коммерческих системах мониторинга социальных медиа. Тем не менее для русскоязычного контента до настоящего времени отсутствовали общедоступные размеченные корпуса, на которых разработчики могли бы провести оценку качества своих методов. Данный пробел были призваны восполнить новые дорожки семинара РОМИП, в рамках которых участникам предлагалось решить задачу классификации отзывов о книгах, фильмах и фотокамерах.

В настоящей работе исследуются методы решения задачи классификации отзывов о книгах 2 (положительный, отрицательный) и 3 (положительный, отрицательный, нейтральный) класса в рамках новых дорожек РОМИП.

Постановка задачи

Участникам была предложена тестовая коллекция, представляющая собой набор отзывов пользователей рекомендательного портала Imhonet.ru на книги различных жанров (всего 24 160 отзывов). Каждый отзыв имел пользовательскую оценку от 1 до 10 баллов. Из имеющихся дорожек нами были выбраны две: дорожка по классификации отзывов пользователей на 2 класса и дорожка по классификации отзывов пользователей на 3 класса. В первом случае требовалось разделить отзывы на положительные и отрицательные. Во втором случае требовалось разделить отзывы на 3 класса: «положительный», «средний» (в отзыве указываются достаточно значимые положительные и отрицательные стороны оцениваемой книги) и «отрицательный».

Среди особенностей задачи следует отметить сильный дисбаланс тестовой коллекции в сторону положительных отзывов.

Формирование обучающей выборки

Двое экспертов независимо оценивали тестовую коллекцию и проставляли оценки: негативный отзыв, позитивный отзыв, отзыв содержит как положительные, так и отрицательные характеристики. Каждый эксперт оценил порядка 4000 отзывов, большая часть из которых была отнесена к положительным. В качестве обучающей выборки в разных прогонах брались как результат оценки одного эксперта, так и множество пересечений оценок обоих экспертов (отзывы, для которых оба эксперта выставили одинаковую оценку). Согласованность оценок экспертов при формировании обучающей выборки достигала 80%.

Представление документа и извлечение терминов

Исследования проводились в рамках векторной модели представления документов, при которой документ описывается набором выделенных из текста терминов. В работе исследуется возможность обогащения классификационных признаков, полученных автоматическим (базовым) методом, который хорошо зарекомендовал себя в задаче тематической классификации документов [1–4], путем добавления к ним терминов, выделенных в рамках лингвистического подхода с использованием словарей оценочной лексики.

В базовом методе в качестве однословных терминов выделялись все слова документа за исключением служебных частей речи, числительных и дат. Многословные термины выделялись при помощи алгоритма синтактико-семантического анализа [5] и представляли собой простые именные группы (напр. «глубокая мысль», «классика жанра»). Именные группы были усложнены включением в их структуру конструкций с предлогами в соответствии с моделями управления [6] (напр. «взгляд на мир», «книга для детей»).

Для повышения качества рубрицирования и обогащения набора классификационных признаков был применен лингвистический подход. Путем анализа имеющихся отзывов эксперт выделил атрибуты книги, на которые большинство пишущих обращали внимание. Таким образом, был получен список наиболее значимых для читателей вещей: язык, сюжет, герои, концовка, впечатления от прочтения, автор и т. д. Список данных атрибутов был расширен синонимами (*книга=книженция=опус=чтиво=произведение=сочинение* и т. д.; *конец=концовка=финал=развязка=хэппи-энд*; *герой=персонаж=характер* и т. д.) и гипонимами (*книга=роман=повесть=рассказ=детектив=пьеса=фэнтези=поэма* и т. д.; *автор=писатель=поэт*).

Далее были составлены словари оценочной лексики (прилагательные и глаголы), выражающей положительную, отрицательную или среднюю оценку. Примеры положительной оценки: *бесподобный, великолепный, яркий; запомниться, нравиться, потрясать*. Примеры отрицательной оценки: *бессодержательный, занудный, пошлый; устареть*. Примеры оценки «средне»: *неоднозначный, неровный, средненький, специфический*.

Для лингвистического анализа текста были использованы семантические шаблоны, описывающие возможные синтаксические связи в предложении между группами терминов из получившихся словарей [7]. Шаблон задает лексико-грамматические ограничения на искомую конфигурацию связей между словами в тексте, которые определяются синтаксическим анализатором. На Рисунке 1 приведен семантический шаблон для извлечения оценки книги, которая выражается прилагательным в конструкциях вида: *Книга оказалась достаточно интересной*; *Эти писатели стали культовыми еще в 60-е годы*.

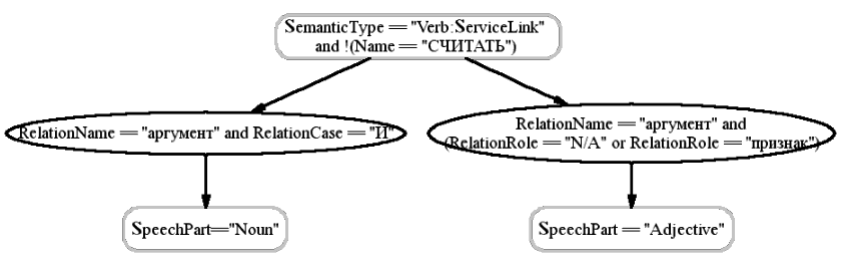


Рис. 1. Пример семантического шаблона для определения оценки книги

В вершинах указываются ограничения на части речи (*SpeechPart="Noun"* — существительное), конкретные слова *(!Name=="СЧИТАТЬ"* — запрещен глагол «считать»), или семантические разряды слов (*SemanticType="Verb:ServiceLink"* — глагол-связка). В эллипсах описываются ограничения на синтактико-семантические связи между словами: тип связи (*RelationName=="аргумент"*), семантическая роль (*RelationRole=="признак"*), падеж (*RelationCase=="И"* — именительный). Окончательно такой шаблон параметризуется множеством конкретных слов из соответствующих словарей: множеством синонимов и гипонимов для книги и ее атрибутов параметризуется узел с ограничениями *SpeechPart="Noun"*,

имеющий роль «Объект оценки» (*EstimatedObject*); множеством оценочных слов параметризуется узел с ограничениями *SpeechPart="Adjective"*, имеющий роль «Оценка» (*QualityAdjective*); узел с ограничением *SemanticType == "Verb:ServiceLink" and !(Name == "СЧИТАТЬ")* параметризуется множеством глаголов-связок за исключением глагола «считать». Выделенные таким образом пары «объект оценки + оценка» использовались в качестве терминов для автоматического определения тональности отзывов. Примеры положительных терминов: *глубокое произведение; книга зацепила; любить книгу; произведение интересное; проглотить книгу; сильная вещь; сюжет захватывает; хорошее чтение; финал неожиданный; язык легкий*. Примеры отрицательных терминов: *дочитать с трудом; испортить настроение; книга бессмысленная; неудачный перевод; скучная книга; сюжет не затягивает; стиль не понравился; читать по диагонали; язык уродливый*.

Методы классификации

В работе исследованы два подхода. В первом подходе для обучения классификатора использовались оценки самих пользователей. По ним строилась линейная регрессионная модель в реализации SVM-Light [8]. Затем по этой модели вычислялись веса документов из обучающей выборки и подбирались пороги отнесения документа к заданным классам таким образом, чтобы получить наилучшее соответствие между получаемым разбиением и разметкой экспертов (максимизировалась F-мера).

Во втором подходе классификатор строился только на основе обучающей выборки, сформированной экспертами. Рассмотрены следующие методы классификации:

- Линейный классификатор, в котором обучение производится для каждого класса независимо от других классов, в реализации SVM-Light [8]. Если в процессе обработки тестовой выборки один документ попадал в несколько классов, то мы принудительно относили его к одному классу, в котором этот документ имел самый большой вес.
- Линейный классификатор, который строит множество непересекающихся классов, то есть ставит в соответствие документу ровно в один из заданных классов. Использовалась реализация SVM-Multiclass [9].
- Линейный классификатор, который обучается независимо на классах положительных и отрицательных отзывов в реализации SVM-Light [8], а используется в задаче классификации на 3 класса. К классу нейтральных отзывов мы относили документы, которые классификатор приписывал одновременно и классу положительных, и классу отрицательных отзывов.

Результаты

В работе проанализированы результаты оценки 18 прогонов классификации на 2 класса и 24 прогона классификации на 3 класса. Прогоны варьировались:

- по способу формирования обучающей выборки: оценка первого эксперта [expert-1], второго [expert-2], согласованная оценка экспертов (обучающая выборка содержала только документы, которые эксперты оценили одинаково) [expert-and];
- по способу извлечения терминов: автоматический [base], смешанный (обогащение базового набора классификационных признаков в рамках лингвистического подхода с использованием словарей оценочной лексики) [hybrid];
- по методу классификации: регрессия [regression], линейный классификатор на независимые классы [one-per-class], на непересекающиеся классы [multiclass], линейный классификатор, обучающийся на классах положительных и отрицательных отзывов, но применяемый в задаче классификации на 3 класса [2-to-3-class].

Влияние различных подходов на качество классификации мы оценивали с помощью F1-меры [10], сильные и слабые требования к релевантности обозначены далее соответственно F-and и F-or.

Согласно Таблице 1, первый эксперт сформировал немного лучшую по качеству обучающую выборку, чем второй эксперт (лучшую в смысле качества обучения исследуемых методов), хотя различие между ними не превосходит нескольких процентов. Здесь и далее «average» и «maximum» обозначают в таблицах способ обобщения результатов по различным прогонам. В первом случае вычисляется средний результат, во втором случае берется максимальное значение. В частности, в Таблице 1 усреднение берется по различным методам и способам отбора терминов.

Таблица 1. Качество работы классификатора, построенного по разным обучающим выборкам, в задаче классификации на 2 и 3 класса

	average		maximum	
2-class	F-and	F-or	F-and	F-or
expert-1	0.691	0.721	0.723	0.747
expert-2	0.669	0.685	0.721	0.732
expert-and	0.690	0.706	0.723	0.724

	average		maximum	
3-class	F-and	F-or	F-and	F-or
expert-1	0.465	0.508	0.536	0.577
expert-2	0.419	0.449	0.484	0.516
expert-and	0.440	0.474	0.521	0.560

Данные, приведенные в Таблице 2, свидетельствуют о том, что привлечение «продвинутых» лингвистических признаков дает незначительное улучшение результата для задачи классификации на 2 класса и даже немного ухудшает результат в случае 3 классов. Небольшой прирост результата можно объяснить тем, что извлекаемые в базовом методе термины оказались, по сути, эквивалентны большинству созданных семантических шаблонов, за исключением шаблонов, содержавших глаголы. Ухудшение результатов для задачи классификации на 3 класса, связано с некорректной группировкой терминов для класса нейтральных отзывов с использованием словарей оценочной лексики. Словарь оценочной лексики для нейтральных отзывов содержал только

слова, соответствующие средней оценке, например, «средненький», «специфический», «сносный», «читабельный», «неровный». Но эта категория содержит согласно постановке задачи помимо нейтральных отзывов также документы, в которых книгу и хвалят, и ругают одновременно. В результате получилась низкая полнота покрытия языкового материала лингвистическими шаблонами документов данной рубрики.

Таблица 2. Качество работы классификатора в зависимости от способа извлечения терминов, описывающих документ, в задаче классификации на 2 и 3 класса

	average		maximum	
2-class	F-and	F-or	F-and	F-or
base	0.679	0.699	0.709	0.727
hybrid	0.688	0.709	0.723	0.747

	average		maximum	
3-class	F-and	F-or	F-and	F-or
base	0.445	0.481	0.536	0.577
hybrid	0.437	0.473	0.512	0.554

Согласно Таблице 3, в задаче классификации на 2 класса наилучший результат был достигнут при помощи метода one-per-class. Остальные методы незначительно ему уступают. В задаче классификации на 3 класса лучшим оказался метод regression. Ему немного уступает one-per-class. Следует отметить, что наименьший разброс результатов при варьировании способов отбора классификационных признаков и формирования обучающей выборки дает метод one-per-class, что свидетельствует о меньшей чувствительности данного метода к качеству входных данных по сравнению с другими методами.

Таблица 3. Качество работы классификатора в зависимости от метода обучения в задаче классификации на 2 и 3 класса

	average		maximum	
2-class	F-and	F-or	F-and	F-or
regression	0.674	**0.715**	0.701	0.727
one-per-class	**0.699**	**0.715**	**0.723**	**0.747**
multiclass	0.677	0.682	**0.723**	0.735
–	–	–	–	–

	average		maximum	
3-class	F-and	F-or	F-and	F-or
regression	**0.494**	**0.529**	**0.536**	**0.577**
one-per-class	0.487	0.525	0.512	0.554
multiclass	0.355	0.375	0.418	0.453
2-to-3-class	0.429	0.479	0.459	0.507

На Рисунках 2 и 3 показаны сводные результаты по всем участникам, участвовавшим в дорожках классификации отзывов о книгах на 2 и 3 класса, соответственно. Наши прогоны обозначены темной заливкой и упорядочены на рисунках по возрастанию меры F-or, прогоны других участников обозначены более светлой заливкой и упорядочены по убыванию F-or. Использование разметки экспертом 1 дало равномерно более высокий результат по сравнению с разметкой эксперта 2 или их согласованной оценкой (возможно, это свидетельствует о том, что эксперт 1 имеет больше опыта). Поэтому наши прогоны на Рисунках 2 и 3 приводятся только для классификаторов, построенных

по обучающей выборке первого эксперта. Расшифровка идентификаторов этих прогонов дана в Таблице 4.

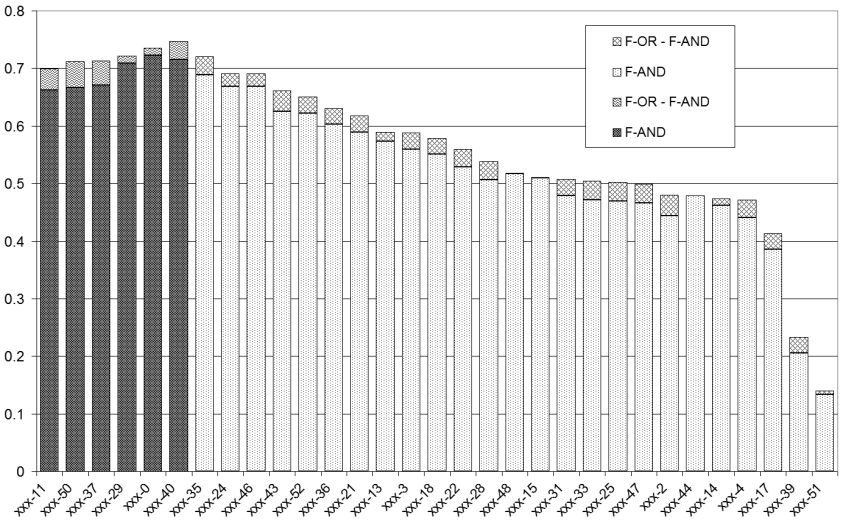


Рис. 2. Результаты оценки прогонов дорожки классификации на 2 класса

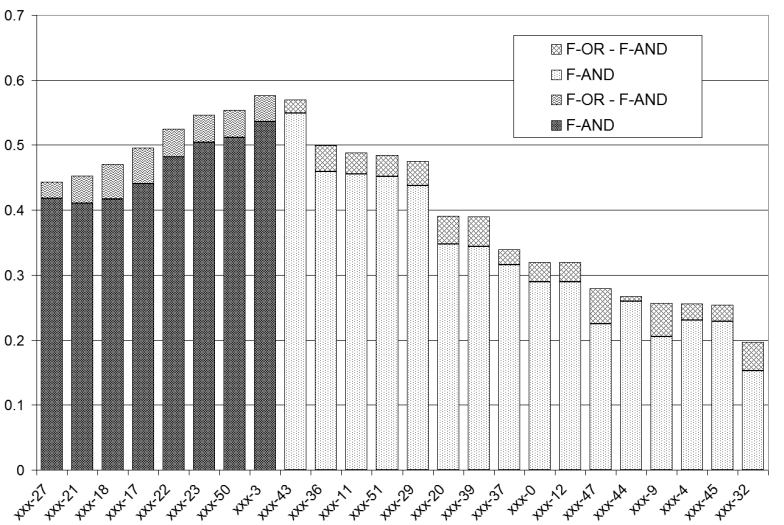


Рис. 3. Результаты оценки прогонов дорожки классификации на 3 класса

Таблица 4. Расшифровка идентификаторов наших прогонов. Номер строки определяет тип метода классификации, номер столбца — способ извлечения классификационных признаков

2-class	base	hybrid
regression	xxx-50	xxx-37
one-per-class	xxx-11	xxx-40
multiclass	xxx-29	xxx-0
–	–	–

3-class	base	hybrid
regression	xxx-3	xxx-23
one-per-class	xxx-22	xxx-50
multiclass	xxx-27	xxx-21
2-to-3-class	xxx-18	xxx-17

Заключение

Проведена апробация ряда методов решения задач классификации отзывов о книгах на 2 и 3 класса. Установлено, что использование методов, обычно применяемых для решения задачи тематической классификации, позволяет получить достаточно высокое качество, сопоставимое с наилучшими результатами дорожек РОМИП по классификации Веб-сайтов и нормативно-правовых документов. Предложен метод обогащения классификационных признаков в рамках лингвистического подхода с применением словарей оценочной лексики, который дает незначительное улучшение результата при классификации на 2 класса. В дальнейшем мы планируем более детально исследовать возможность применения экспертно-лингвистических подходов для построения классификационных признаков.

Литература

1. *Плешко В. В., Ермаков А. Е., Голенков В. П., Поляков П. Ю.* RCO на РОМИП 2005 // Труды третьего российского семинара РОМИП'2005. (Ярославль, 6 октября 2005г.). — Санкт-Петербург: НИИ Химии СПбГУ — 2005 — с. 106–124.
2. *Поляков П. Ю., Плешко В. В.*, RCO на РОМИП 2006 // Труды четвертого российского семинара РОМИП'2006. (Суздаль, 19 октября 2006г.). — Санкт-Петербург: НУ ЦСИ — 2006 — с. 72–79.
3. *Плешко В. В., Поляков П. Ю.*, RCO на РОМИП 2008 // Труды РОМИП 2007–2008. (Дубна, 9 октября 2008г.). — Санкт-Петербург: НУ ЦСИ, 2008 — с. 96–107.
4. *Плешко В. В., Поляков П. Ю., Ермаков А. Е.* RCO на РОМИП 2009 // Труды РОМИП 2009. (Петрозаводск, 2009г.). — Санкт-Петербург: НУ ЦСИ, 2009 — с. 122–134.
5. *Ермаков А. Е.* Значимость элементов текста в свете теории синтаксической парадигмы // Русский язык: исторические судьбы и современность.

II Международный конгресс исследователей русского языка. Труды и материалы. — Москва: МГУ — 2004.

6. *Ермаков А. Е.* Эксплицирование элементов смысла текста средствами синтаксического анализа-синтеза // Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог'2003 (Протвино, 11–16 июня, 2003 г.). — Москва, Наука, 2003
7. *Ермаков А. Е.* Извлечение знаний из текста и их обработка: состояние и перспективы // Информационные технологии. — 2009. — N 7.
8. *Joachims T.* Making large-scale support vector machine learning practical // Advances in Kernel Methods: Support Vector Machines / B. Scholkopf, C. Burges, A. Smola (eds.) — MIT Press: Cambridge, MA" — 1998.
9. *Joachims T., Finley T., Yu Chun-Nam.* Cutting-Plane Training of Structural SVMs // Machine Learning Journal. — 2009, V.77, No.1, pp.27–59.
10. *Chetviorkin I., Braslavskiy P., Loukachevitch N.* Sentiment Analysis Track at ROMIP 2011.

PROOF OF CONCEPT STATISTICAL SENTIMENT CLASSIFICATION AT ROMIP 2011

Poroshin V. (vladimir.poroshin@m-brain.com)

M-Brain Oy, Helsinki, Finland

In this paper we present a simple statistical classification method that predicts whether the opinion expressed by text in natural language is positive or negative. There are two main approaches in the sentiment or opinion detection: linguistic rule based systems and statistical algorithms. While statistical methods are easier to build when sufficient training data is available, it is widely perceived that a linguistic system can deliver better results. Our work was intended to prove the concept that a simple Naïve Bayes based statistical classification algorithm with a minor language dependent adaptation is able to perform well in a binary sentiment classification task. In order to prove the hypothesis, we participated in Russian Information Retrieval Seminar (ROMIP) 2011 sentiment classification track [1], and achieved quite competitive results in sentiment prediction of Russian blog posts. This paper contains a detailed description of our classification method, including a feature extraction and normalization process, training and test data, evaluation metrics; and presents our official ROMIP results.

Keywords: statistical sentiment classification, sentiment analysis, sentiment detection, opinion mining, Naïve Bayes, ROMIP

1. Introduction

The goal of this work is to compare a simple statistical based approach for a sentiment classification problem to other statistical and linguistic methods in the scope of the ROMIP 2011 sentiment classification track [1]. The task of the sentiment analysis in our context lies in automatic categorization of incoming text in Russian into two classes: positive or negative in general, i. e. without any specified target of the sentiment. This is one variation of the sentiment classification problems, which in general can vary in number of classes to predict (2, 3, 5 or more classes, including neutral and other emotional states such as angriness, sadness and so on) or in the target of the sentiment (specific word, sentence, whole text, etc.). Although, a binary 'no target' sentiment classification in many cases is simpler than the other sentiment classification tasks; it can serve as a basis for implementation of some of them.

Sentiment analysis can be viewed as a classification problem where one can use well-known statistical classifiers, as it is outlined in a number of publications [2, 4]. Our research aims to show that a simple statistical classifier with a pretty generic feature extraction process can achieve good results in the sentiment classification.

The paper is organized in the following way: section 2 contains the review of a modified Naïve Bayes classifier; section 3 describes features and their extraction process; section 4 illustrates test and training data; section 5 presents our official ROMIP 2011 evaluation results; and, finally, section 6 completes the paper with conclusions.

2. Method description

In order to assign sentiment labels to new test documents we use Naïve Bayes algorithm with few modifications as our classification method.

In multinomial Naïve Bayes [5] a class C is assigned to a test document d, where

$$C = argmax_c P_{NB}(c|d)$$

$$P_{NB}(c \mid d) = \frac{P(c)\sum_{i=1}^{m} P(f \mid c)^{n_i(d)}}{P(d)} \quad (1)$$

where f is one of m features, $n_i(d)$ is the count of feature f_i in a document d. $P(c)$ and $P(f|c)$ are calculated through maximum likelihood estimates, and an add-1 smoothing is utilized for unseen features.

With the focus on the improvement of a standard multinomial Naïve Bayes we applied the following modifications: a term frequency (TF) transformation (1.1) and a TF transformation based on length (1.2).

In a TF transformation all term frequencies $n_i(d)$ in the formula (1) are replaced by:

$$n_i(d) = \log(n_i(d)+1) \quad (1.1)$$

It has been shown in [3] that this transformation models the term distribution of the text in a better way by reducing the weight of frequent features.

Other TF transformation normalizes feature counts to deal with the negative effect of long documents, since Naïve Bayes assumes independence of features [3]. For this we transform counts $n_i(d)$ to

$$n_i(d) = \frac{n_i(d)}{\sqrt{\sum_k n_k(d)^2}} \quad (1.2)$$

3. Features

We have tried several types of features, including words uni-grams, bi-grams, word-forms, stems and lemmas. The best found combination was to use uni- and bi-grams of lemmatized words. Stop-words were also removed from the text.

The process of lemmatization lies in the determination of a dictionary form (or lemma) of a given word. In many languages, including English and Russian,

a word-form can have more than one lemma when it is considered independently from the context. For instance, a Russian word form *моя* can be lemmatized to a possessive pronoun *мой* (*mine*) or to a verb *мыть* (*to wash)*. Usually, selection of the correct lemma can be deduced from the context sentence with the help of the part of speech tagger or some other linguistic processing. In our case we use a frequency dictionary to select the most probable lemma. In case lemmas of some word-form are not presented in the dictionary, random ones are picked.

In terms of the nature of the sentiment classification task, we have found that weight of negative particles (such as *not* in English) in our method usually dominates in case they are present. For instance, a phrase "*it is not bad*" will be classified as negative, because words *not* and *bad* have quite high frequencies in the training data with negative sentiment labels. As a result many test documents can be incorrectly classified as negative ones only because they contain a negative particle. To overcome this problem we used a data pre-processing step that glues a negative particle to the word next to it. In our example, it will become "*it is notbad*".

Our full feature extraction process is the following:

- replace all URLs to a special label *tokenurl*
- replace all positive and negative emoticons to special labels *tokensmilepositive* and *tokensmilenegative* correspondingly
- lowercase all text
- remove repeated letters
- remove stop words
- glue negative particles as it was described
- lemmatize each word
- collect uni- and bi-grams as features

"Remove repeated letters" step becomes quite necessary when we deal with the text from social media sources, for example, from Twitter. There the words are usually misspelled by repeating the letters, for example, "*woooo!!!! Suuuch a messsss! brrrrr....*". During this step we delete all letters that appear in a word more than 2 times, i. e. in our example the saying will be transformed to "*woo!!!! Suuch a mess! brr....*".

4. Test and Training data

Our training data was collected from three sources: the web site http://lovehate.ru, Yandex market http://market.yandex.ru and Twitter.

Main portion of the training data was collected from the web site lovehate.ru, which contains opinions in Russian on various topics. There people mark themselves their comments on some topics as positive or negative.

From Twitter we got a sample of positive and negative tweets in Russian by collecting tweets from the Twitter API with emoticons as a query. A tweet with a negative emoticon, such as :(:-(is considered to have a negative sentiment, and a tweet with a positive emoticon — a positive one [4]. We used only a small portion of such data

for training of the classifier because of the low quality of it. The decision to include the Twitter training data was connected with intention to have internet slang words in our model.

The last set of the training data is a dump of positive and negative opinions about digital cameras in corresponding Yandex market web pages [6]. This data was received from ROMIP as a part of the in-domain training set, since one of the ROMIP 2011evaluation topics in the sentiment classification track was about digital camera products.

We did not use other official ROMIP 2011training data.

A summary of the training data sources is presented in the table 1.

Table 1. Training data

	Number of topics	Total number of words	Total number of samples	Is it in-domain?
lovehate.ru	2850	20267645	346041	No
ROMIP Yandex market digitalcam	1	602101	19986	Yes
Twitter	N/A	2527064	63511	No

As participants of the ROMIP 2011 we also received a test data, which includes a set of blog posts on 3 topics: digital cameras', books' and movies' reviews. In the evaluation of our method we considered only digital cameras testing set, because we used a corresponded in-domain training data only for this topic. Test documents were manually judged by two assessors in order to create a human quality sentiment labels for evaluation. For simplicity we used evaluation results calculated only for test samples where both assessors assigned the same labels (table 2).

5. Evaluation results

Official ROMIP evaluation results of our algorithm (denoted as *stats*) are shown in the table 2. Based on average F-Measure score the proposed statistical method is on the 4th place out of 25 total results. Average of all participants' results is also presented in the table 2.

Table 2. Official ROMIP 2011 results for 2 class sentiment classification track for the digital cameras topic (first 5 out of 25 total runs by participants sorted by F-Measure_AND ***F*** and an average over all runs)

	P	R	F	A	P_P	P_N	R_P	P_N	F_P	F_N
xxx-24	0.9092	0.9337	0.9209	0.9569	0.9811	0.8372	0.9674	0.9	0.9742	0.8675
xxx-9	0.8905	0.9291	0.9082	0.9490	0.9810	0.8	0.9581	0.9	0.9694	0.8471
xxx-16	0.9355	0.88052	0.9052	0.9529	0.9593	0.9118	0.9860	0.775	0.9725	0.8378

	P	R	F	A	P_P	P_N	R_P	P_N	F_P	F_N
stats	**0.8562**	**0.8416**	**0.8486**	**0.9216**	**0.9493**	**0.7632**	**0.9581**	**0.7250**	**0.9537**	**0.7436**
xxx-6	0.8059	0.8808	0.8356	0.9020	0.9703	0.6415	0.9116	0.85	0.9400	0.7312
average	0.7467	0.7692	0.72156	0.81522	0.94716	0.54615	0.83134	0.707	0.8741	0.5690

Average precision P, recall R and F-measure F are calculated as:

$$P = \frac{P_N + P_P}{2}, \quad R = \frac{R_N + R_P}{2}, \quad F = \frac{F_N + F_P}{2}$$

Where precision, recall and F-measure for a positive class:

$$P_P = \frac{tp}{tp + fp}, \quad R_P = \frac{tp}{tp + fn}, \quad F_P = 2\,\frac{P_P \cdot R_P}{P_P + R_P}, \quad \text{where}$$

tp — number of true positives, fp — number of false positives, fn — number of false negatives.

And for a negative class:

$$P_N = \frac{tn}{tn + fn}, \quad R_N = \frac{tn}{tn + fp}, \quad F_N = 2\,\frac{P_N \cdot R_N}{P_N + R_N}, \quad \text{where}$$

tn — number of true negative

Total accuracy of the method is:

$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

6. Conclusion

We presented a statistical classifier for a two class sentiment classification task. Our method is based on slightly modified Naïve Bayes classification algorithm and a simple linguistic data pre-processing, which helps to better suit the domain of the problem.

Our participation in ROMIP 2011 sentiment classification track serves as the evaluation of the proposed method. The results show that our method is competitive enough in comparison to other participants' approaches. This means that even a quite simple statistical method can show good performance in this type of tasks.

In the future perspective, our algorithm can be extended and further improved in several different ways. Some of the domain adaptation techniques such as mixture of models can be used to achieve better results for the data with a predefined topic. Feature extraction process could be also improved by employing a proper stemming technique, which is able to resolve words' ambiguity. Also, other classification models like SVN may perform better in this problem because they don't assume independence of features and better model the data.

The idea to glue negative particles that was described in section 3 also needs an improvement. Not an every word next to negative particle is a target of gluing.

There could be words in between, for example, "it is *not* so *bad*". Employing word dependencies from a syntactical parser will help to find correct targets.

Solution to the two class sentiment problem (positive/negative) can be as well further embedded in a framework, where a neutral class is detected also, for example, with the help of another classifier, which detects neutrality of the text.

References

1. *ROMIP*: Russian Information Retrieval Evaluation Seminar, http://romip.ru
2. *B. Pang, L. Lee, and S. Vaithyanathan*. Thumbs up? Sentiment classication using machine learning techniques. [Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2002], pp. 79–86
3. *Jason D. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger.* Tackling the Poor Assumptions of Naive Bayes Text Classifiers. [Proc. of ICML'2003]. pp.616~623
4. *Go, A., R. Bhayani, and L. Huang.* Twitter sentiment classification using distant supervision. Technical report, Stanford Digital Library Technologies Project 2009
5. *C. D. Manning and H. Schutze* (1999). Foundations of statistical natural language processing. MIT Press
6. *Yandex* Market: http://market.yandex.ru/

КЛАССИФИКАЦИЯ ОТЗЫВОВ ПОЛЬЗОВАТЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ ФРАГМЕНТНЫХ ПРАВИЛ

Васильев В. Г. (vvg_2000@mail.ru),
Худякова М. В. (mariya.kh@gmail.com),
Давыдов С. (davydov_sergey@hotmail.com)

ООО «ЛАН-ПРОЕКТ», Москва, Россия

В работе рассматривается подход к анализу отзывов пользователей, основанный на задании правил классификация и выделения значимых фрагментов на специальном языке. Проводится анализ эффективности автоматического построения и коррекции правил путем обучения на примерах. Приводятся результаты экспериментов в рамках соответствующей дорожки РОМИП 2011.

Ключевые слова: анализ отзывов пользователей, классификация, фрагментные правила

1. Введение

В настоящее время в связи с активным развитием социальных сетей, форумов и блогов вопросы автоматизации анализа мнений пользователей сети по различным вопросам (отношение к товарам и услугам, событиям, высказываниям, сообщениям) вызывают большой интерес у многих организаций, что приводит к активизации научных исследований и экспериментов в данной области. Обычно задача анализа мнений пользователей ставится как задача классификации текстов на два или более класса, которые разделяют мнения на позитивные и негативные, а также их оттенки.

В работе [1] рассматриваются подходы к классификации отзывов на фильмы, основанные на сравнении числа положительных и отрицательных слов с учетом усиливающих терминов, а также использовании стандартного классификатора на основе машин опорных векторов. Как показывают проведенные авторами эксперименты при использовании первого подхода F-мера достигается порядка 60%–70%, а при использовании второго подхода порядка F-мера порядка 80%–85%. В работе [2] также как и в предыдущей работе рассматривается использование словарного подхода и обучения на примерах с использованием SVM. При этом реализована итерационная процедура пополнения словарей положительных и отрицательных терминов за счет классификации неразмеченных текстов. В целом эксперименты на массиве отзывов о различных товарах на китайском языке достигаются значения F-меры порядка 85%–90%. В работе [3] для классификации отзывов все предложения

(высказывания) в тексте предварительно разбивают на личные и нейтральные и осуществляют построение трех классификаторов, которые обучаются на личных, нейтральных и всех предложениям с использованием метода SVM. Как показывают авторы такой подход позволяет несколько повысить качество по сравнению с базовым уровнем. В работе [4] приводится пример построения кросс языкового классификатора для анализа отзывов пользователей. Обучающая выборка представлена на английском языке, а обрабатываются переводы отзывов с китайского языка. Для обучения классификатора используется метод SVM и процедура использования неразмеченных текстов. В целом на отзывах о различных цифровых устройствах авторами были получены значения F-меры порядка 75 %–80 %. В работе [5] в отличие от предыдущих рассмотренных работ рассматривается задача классификации не отзывов о товарах, а мнений политиков о поправках к законам и результатов голосования. Помимо словарного и векторного подхода (метод SVM) для анализа отзывов в ряде работ строятся специальные вероятностные модели. Например, в [6] учитывается дерево синтаксического разбора предложений и зависимости между словами, а в работе [7] строится совместная тематико-оценочная вероятностная модель. Также в ряде работ авторы явно задают правила оценки текстов. В частности, в работе формулируются различные правила для определения области действия инверсных слов типа «не».

Таким образом, в работах по классификации отзывов применяются как стандартные методы классификации текстов, так и модифицированные методы, в которых учитывается возможная инверсия значений оценочных слов, синтаксическая структура предложений, зависимости между словами [6]. Целью настоящей работы является исследование эффективности использования стандартных методов классификации текстов основанных на задании правил и обучении на примерах применительно к задаче классификации отзывов на русском языке, а также определение перспективных направлений совершенствования и развития данных алгоритмов. При этом в качестве основного рассматривается подход к классификации отзывов на основе правил, сформированных экспертами. Оценка эффективности рассматриваемых методов производится в рамки дорожки классификации отзывов пользователей на два класса.

2. Описание используемых подходов

2.1. Классификация на основе правил

Для задания правил в данной работе применяется подход, описанный в работе [8]. В данном случае оцениваемый текст D рассматривается как последовательность элементов (слов, цифр, знаков препинания), т. е. $D = (d_1, ..., d_n)$, где $d_i \in T$ — отдельный элемент текста, $T = (t_1, ..., t_m)$ — множество всех допустимых элементов, n — длина текста, m — число различных допустимых элементов текстов.

Множество $\mathbb{F} = \{(p, q) \mid 1 \le p \le q \le n\}$ будем называть множеством всех фрагментов текста длины n. Фрагментами текста будем называть отдельные элементы данного множества $f = (f_1, f_r) \in \mathbb{F}$, которые задают левую f_1 и правую f_r границы фрагмента (номер начального и конечного элемента текста). Результатом выполнения произвольного правила Q для текста D является множество $F_Q \subset \mathbb{F}$, содержащее все фрагменты удовлетворяющие правилу Q. При этом, если $F_Q \neq \emptyset$, то будем говорить, что текст D удовлетворяет правилу Q.

Операции для задания правил можно разбить на следующие группы:

- элементарные — выделяют фрагменты, соответствующих отдельным словам;
- сложные — выделяют сложные многословных выражений;
- определяющие — задание общих понятий и множеств;
- управляющие — задают параметры классификации и обучения на примерах.

Элементарные операции — выделяют отдельные слова в тексте, предложения, строки, разделы документа. Например, правило $FirstUp — выделяет все слова в тексте с большой буквы, правило Липецкая — все слова, являющиеся словоформами слова «липецкая», правило 'обл*' — все слова начинающиеся на «обл»; $Sentence — все предложения в документе, #section — раздел документа с определенным именем (например, заголовок).

Сложные операции — задания преобразования множеств фрагментов. Приведем примеры определения отдельных операций для построения сложного правила Q на основе правил $Q_1, \ldots, Q_k$.

$Q = Q_1 \nabla Q_2$ — бинарная операция ИЛИ, $F_Q \equiv R(F_{Q_1} \nabla^* F_{Q_2})$, $F_{Q_1} \nabla^* F_{Q_2} = \{f \in \mathbb{F} | \exists f_1 \in F_{Q_1}, f \sqsupset f_1$ или $\exists f_2 \in F_{Q_2}, f \sqsupset f_2\}$. Например, правило *искажение блеклый неуклюжий тьфу* выделяет фрагменты, равные соответствующие отдельным словам.

$Q = Q_1 \Delta_{n_1} Q_2$ — бинарная операция И с ограничением на расстояние между фрагментами, $F_Q \equiv R(F_{Q_1} \Delta^*_{n_1} F_{Q_2})$, $F_{Q_1} \Delta^*_{n_1} F_{Q_2} = \{f \in \mathbb{F} | \exists f_1 \in F_{Q_1}$ и $\exists f_2 \in F_{Q_2}$, т. что $f \sqsupset f_1$, $f \sqsupset f_2$ и $d(f_1, f_2) \le n_1\}$. Например, правило *смазанные &3 образы*, выделяет фрагменты, где расстояние между «смазанный» и «образ» не более 3 слов.

$Q = Q_1 \square_{n_1, n_2} Q_2$ — бинарная операция последовательности с ограничением на расстояние между фрагментами, $F_Q \equiv R(F_{Q_1} \square^*_{n_1, n_1} F_{Q_2})$, $F_{Q_1} \square^*_{n_1, n_1} F_{Q_2} = \{f \in \mathbb{F} | \exists f_1 \in F_{Q_1}$ и $\exists f_2 \in F_{Q_2}$, т. что $f_1 < f_2$, $d(f_1, f_2) > 0$, $f \sqsupset f_1$, $f \sqsupset f_2$ и $n_1 \le d(f_1, f_2) \le n_2\}$. Например, *отказаться :3 (снимать производство)*, выделяет фрагменты, в которых после «отказаться» на расстоянии 3 слов находятся слова «снимать» или «производство».

$Q = \bowtie(Q_1, \ldots, Q_k)$ — множественная операция последовательности соседних элементов (осуществляет отбор смежных фрагментов), $F_Q \equiv R(\bowtie^*(F_{Q_1}, \ldots, F_{Q_k}))$, $\bowtie^*(F_{Q_1}, \ldots, F_{Q_k}) = \{f \in \mathbb{F} | \exists f_i \in F_{Q_1}$, $i=1, \ldots k$, т. что $f_i < f_{i+1}$, $d(f_i, f_{i+1}) = 1$ для $i=1, \ldots, k-1$ и $f \sqsupset f_i$ для $i=1, \ldots, k\}$. Например, правило «(начальник руководитель директор) («главное управление» управление организация отдел) (МВД МЧС МинФин)» — выделяет словосочетания соответствующие руководителям различных ведомств.

$Q = Q_1 \lozenge Q_2$ — бинарная операция нахождения пересечения фрагментов, $F_Q \equiv \{f \in \mathbb{F} | \exists f_i \in F_{Q_1} \wedge f \in F_{Q_2}\}$. Например, правило [великая $FirstUp] — выделяет слова «великая», которые написаны с большой буквы.

$Q = Q_1 \lhd_{n_1,n_2}$ — унарная операция ограничения длины фрагмента, $F_Q \equiv \{f \in F_{Q_1} | n_1 \leq |f| \leq n_2\}$ Например, правило (Нижегородская & Владимирская) #IN #INTERVAL(2w/3w) — выделяет фрагменты, содержащие заданные слова длиной от 2 до 3 слов.

Для возможности построения правил включающих отрицания и условные операторы (наличие выражения проверяется, но оно не включается в итоговый фрагмент) используются специальные варианты бинарных правил, в которых один из операндов считается отрицательным или условным. В частности, символом $\square^{\not\rightarrow}_{n_1,n_2}$ обозначается операция нахождения последовательности, в которой второй операнд берется с отрицанием, символом $\square^{\not\leftarrow}_{n_1,n_2}$ — операция, в которой первый операнд берется с отрицанием, $\square^{\leftarrow}_{n_1,n_2}$ — операция, в которой первый операнд является условным. Определение $\square^{\rightarrow}_{n_1,n_2}$ имеет следующий вид $Q = Q_1 \square^{\rightarrow}_{n_1,n_2} Q_2$, где $F_Q \equiv \{f \in F_{Q_1} | \ \exists! f_2 \in F_{Q_2}$ т. что $f < f_2$, $0 < n_1 \leq d(f,f_2) \leq n_2\}$.

Например, операция *без ^:3 отличн** — выделяет слова, начинающиеся на «отличн» перед которыми нет слова «без».

Определяющие операции –задают понятия в форме шаблонных подстановок (#define) и в форме сохраненных множеств фрагментов (#set). Для обращения к подстановке и множеству фрагментов используются операторы @ и @@. Например, операция

#define Bad плохой глупый

задает понятие Bad, к которому можно обращаться из текста правила с помощью выражения @Bad.

Управляющие операции — задают параметры классификации и обучения. Например, операция #option train задает необходимость автоматического формирования правил путем обучения на примерах.

Для классификации отзывов был разработан набор понятий, из которых были сформированы правила для выделения положительных и отрицательных отзывов. Например, правило для определения отрицательных отзывов имеет следующий вид

@CheckBadBegin
@Bad & ^ (@CheckGoodBegin @CheckBadBegin @Good)

Оно работает следующим образом, сначала проверяются первые два предложения на содержание оценочных слов с использованием правила @ *CheckBadBegin,* а затем проверяется наличие отрицательных слов при условии, что не найдены в начале текста положительные или отрицательные оценки.

Правило проверки на отрицательность начала текста проверяет, что в первых двух предложениях от начала документа перед отрицательным

фрагментом нет положительного фрагмента и слова «не» или двойных кавычек. При этом понятие @@*isbad* является более строгим, а понятие @@*badmark* менее строгим.

Правило для проверки на отрицательность текста целиком @Bad является более сложным, но в целом похожим на @CheckBadBegin. Основу правил составляют понятия *@@isgood, @@isbad, @@goodmark, @@badmark* , которые выделяют множества исходных положительных и отрицательных фрагментов без учета модификаторов перед ними. Определение каждого такого понятия включает около сотни выражений.

В целом алгоритм оценки отзыва при использовании построенных правил имеет следующий вид.

Алгоритм 1. Классификация отзывов на основе правил

Шаг 1. Проверка начала текста, если решение однозначно, то завершить работу.

Шаг 2. Проверка текста в целом, если решение однозначно, то завершить работу.

Шаг 3. Вычисление веса положительных и отрицательных фрагментов;

Шаг 4. Отнесение текста к классу с наибольшим весом.

Построение правил классификации вручную является достаточно трудоемкой процедурой. По этой причине в используемом языке имеются операции, которые позволяют уточнить ранее построенное правило путем анализа результатов классификации обучающей подборки документов. В частности, правило для классификации отрицательных фрагментов было изменено следующим образом.

```
@CheckBadBegin
(@Bad @AutoSupplementQuery) & ^ (@CheckGoodBegin @CheckBadBegin)

#define AutoSupplementQuery $True
```

В приведенном правиле понятие *@AutoSupplementQuery* вычисляется автоматически таким образом, чтобы максимально повысить полноту правила, без снижения точности. Для формирования данного правила используется модифицированный вариант жадного алгоритма построения решающего списка [9], в котором в качестве множества положительных примеров используются неправильно классифицированные отрицательные тексты, а в качестве множества отрицательных примеров все положительные тексты.

Формирование обновленного правила происходит в соответствии со следующей схемой.

Алгоритм 2. Формирование обновленного правила

Шаг 1. Выполнить классификацию обучающего множества с помощью правила, в котором *@AutoSupplementQuery* не задан.

Шаг 2. Выполнить оценку качества классификации и построить @ *AutoSupplementQuery* с использованием модифицированного варианта жадного алгоритма построения решающего списка.

Шаг 3. Выполнить дополнительную коррекцию построенного правила экспертом.

После построения модифицированного правила классификация отзывов происходит с использованием алгоритма 1.

2.2. Классификация с использованием обучаемых алгоритмов

В настоящее время разработано большое количество алгоритмов машинного обучения для решения задач классификации текстов. В данной работе был решено провести тестирование следующих стандартных алгоритмов [10]:

- алгоритм к-ближайших соседей;
- алгоритм построение деревьев решений C4.5;
- алгоритм на основе машин опорных векторов;
- байесовский классификатор на основе смеси многомерных нормальных распределений;
- байесовский классификатор на основе смеси распределений фон Мизеса-Фишера;
- центроидный классификатор Роччио.

Общая схема алгоритма обучения в данном случае является достаточно стандартной и имеет следующий вид.

Алгоритм 3. Обучение классификатора на примерах

1. Формирование векторного представления текстов в рамках модели «Bag Of Words».
2. Снижение размерности (селекция признаков по частоте) и вычисление весов признаков (TF_IDF).
3. Обучение и оценка классификатора на обучающей выборке с использование 5-шаговой процедуры кросс-проверки.

3. Эксперименты

3.1. Описание тестовых массивов и показателей качества

В данной работе эксперименты по оценке качества проводились в рамках дорожки РОМИП 2011 классификации отзывов на два класса. Данная дорожка содержала три обучающих массива текстов:

- массив отзывов о фильмах — содержит 15 718 текстов, предоставленных онлайновой службы рекомендаций IMHONET, каждый отзыв оценен по 10 бальной шкале;
- массив отзывов о книгах — содержит 24 159 текстов, предоставленных онлайновой службы рекомендаций IMHONET, каждый отзыв оценен по 10 бальной шкале;
- массив отзывов о цифровых фотоаппаратах — содержит 10 370 текстов, предоставленных Yandex, каждый отзыв оценен по 5 бальной шкале.

Для тестирования использовался набор из 16 821 текстов, содержащих описание различных объектов интереса пользователей. Задачей дорожки было отнести каждый текст к классу положительных, либо к классу отрицательных отзывов.

Для оценки качества работы классификаторов в настоящей работе использовались следующие стандартные показатели качества: точность, полнота, F1-мера, аккуратность и среднее евклидово расстояние. Для первых трех показателей вычислялись значения, как для отдельных классов, так и макро-оценки.

3.2. Результаты экспериментов

Эксперименты проводились в два этапа. На первом этапе была выполнена самооценка качества классификации с использованием обучающего множества текстов, предоставленного организаторами дорожки. На втором этапе была выполнена обработка тестового множества текстов с использованием отдельных классификаторов и получены оценки качества от организаторов дорожки.

В следующей таблице приведены результаты самооценки качества, полученные с использованием классификаторов на основе правил. При этом классификатор на основе ручных правил обозначен Q1, а классификатор на основе обученных правил Q2.

Таблица 1. Результаты самооценки качества классификации с использованием правил

Классификатор	Объект	Точность положительные	Полнота положительные	Точность отрицательные	Полнота отрицательные
Q1	book	65%	66%	85%	43%
Q1	camera	71%	86%	83%	77%
Q1	film	60%	64%	71%	35%
Q2	book	71%	62%	84%	56%
Q2	camera	69%	88%	83%	81%
Q2	film	61%	67%	72%	37%

Как можно заметить из приведенной таблицы 1 использование процедуры обучения повышает полноту классификации на обучающем множестве, но при этом снижает немного точность.

Также были проведены эксперименты по оценке качества работу обучающих алгоритмов. В следующей таблице, в качестве примера, приведены показатели качества для массива отзывов о книгах. В таблице 2 используются следующие обозначения алгоритмов: SVM — классификатор машин опорных векторов, GMM — байесовский классификатор на основе смеси многомерных нормальных распределений, ROC — классификатор Роччио, KNN — классификатор к-ближайших соседей, VMF — классификатор фон Мизеса-Фишера, TREE — классификатор на основе деревьев решений.

Таблица 2. Результаты оценки качества для массива отзывов о книгах

Классификатор	Объект	Точность положительные	Полнота положительные	Точность отрицательные	Полнота отрицательные
SVM	book	86%	99%	41%	44%
GMM	book	88%	73%	27%	42%
ROC	book	92%	18%	27%	8%
KNN	book	87%	78%	23%	30%
VMF	book	94%	47%	31%	57%
TREE	book	90%	70%	27%	30%

Как можно заметить из приведенной таблицы показатели качества для отрицательных текстов при использовании обучающих алгоритмов заметно хуже, чем при классификации на основе правил. При этом наиболее высокие показатели продемонстрировали алгоритмы: SVM, KNN, TREE. Алгоритм SVM и так достаточно часто используется в различных работах, по этой причине было решено отправить организаторам конкурса результаты обработки тестового массива с помощью алгоритмов KNN и TREE.

Организаторами дорожки для уменьшения субъективности оценок экспертов были рассмотрены 2 схемы оценок качества:

- схема И — учитываются только те отзывы, для которых совпадают оценки экспертов.
- схема ИЛИ — ответ алгоритма считается правильным, если он совпадает с ответом одного из экспертов.

Результаты экспериментов по каждой схеме приведены в следующих двух таблицах. В данные таблицы включена наилучшая оценка по дорожке и результаты оценки качества для 4 прогонов: Q1 — классификатор на основе правил, Q2 — модифицированный классификатор на основе правил, Q3 — классификатор на основе деревьев решений, Q4 — классификатор к-ближайших соседей.

Таблица 3. Результаты оценки качества в соответствии со схемой И

Метод	Объект	Макро-Точность	Макро-Полнота	Макро-F1
Q1	book	0.53	0.58	0.53
Q2	**book**	**0.55**	**0.66**	**0.58**
Q3	book	0.52	0.54	0.53
Q4	book	0.54	0.51	0.51
xxx-20	**book**	**0.96**	**0.61**	**0.67**
Baseline	book	0.46	0.5	0.48
Q1	**camera**	**0.81**	**0.88**	**0.84**
Q2	camera	0.79	0.87	0.83
Q3	camera	0.50	0.47	0.48
Q4	camera	**0.93**	0.54	0.53
xxx-24	**camera**	**0.91**	**0.93**	**0.92**
Baseline	camera	0.42	0.5	0.45
Q1	**film**	**0.67**	**0.70**	**0.68**
Q2	film	0.66	0.70	0.68
Q3	film	0.54	0.53	0.50
Q4	film	0.54	0.52	0.52
xxx-23	**film**	**0.76**	**0.78**	**0.77**
Baseline	film	0.42	0.5	0.45

Таблица 4. Результаты оценки качества в соответствии со схемой ИЛИ

Метод	Объект	Макро-Точность	Макро-Полнота	Макро-F1
q1	book	0.56	0.62	0.56
q2	book	**0.57**	**0.69**	**0.61**
q3	book	0.52	0.55	0.47
q4	book	0.54	0.51	0.51
xxx-20	book	**0.73**	**0.74**	**0.73**
Baseline	book	0.46	0.5	0.48
q1	**camera**	**0.83**	**0.90**	**0.86**
q2	camera	0.83	0.89	0.85
q3	camera	0.53	0.52	0.51
q4	camera	**0.93**	0.54	0.53
xxx-24	camera	**0.92**	**0.94**	**0.93**
Baseline	camera	0.43	0.5	0.48
q1	film	**0.69**	**0.73**	**0.71**
q2	film	0.68	0.73	0.70

Метод	Объект	Макро-Точность	Макро-Полнота	Макро-F1
q3	film	0.56	0.57	0.53
q4	film	0.54	0.53	0.53
xxx-23	film	**0.78**	**0.80**	**0.79**
Baseline	film	0.42	0.5	0.46

Анализ результатов, приведенных в таблицах 3 и 4, позволяет сделать следующие выводы. Методы классификации на основе правил показали более высокое качество работы. Значительно лучше обрабатывается массив с камерами, что связано с тем, что первоначальная настройка правил делалась именно на нем. Обучаемые методы показали низкое качество работы возможно по следующим причинам: учитывались все признаки в текстах, не учитывался контекст употребления слов, методы на основе деревьев решений и классификатор ближайших соседей на обучающей выборке работали хуже метода SVM.

4. Выводы

Таким образом, в настоящей работе рассмотрены несколько подходов к классификации отзывов пользователей. Наиболее эффективным оказался подход, основанный на ручном построении правил экспертами. Использование традиционных методов обучения на примерах, а также расширения запросов с помощью отдельных терминов не приводит к высокому качеству классификации. Это связано, по-видимому, с тем, что в стандартных методах используется теоретико-множественная модель текстов, в которой не учитывается контекст употребления слов.

В качестве перспективных направлений дальнейших исследований можно сформулировать следующие: реализация специальных обучающих алгоритмов для формирования контекстных правил для заданных пользователем оцениваемых объектов, что позволит значительно снизить трудоемкость формирования правил экспертами; выполнение обучения классификаторов не на полных текстах, а на отдельных предложениях, содержащих ссылки на оцениваемый объект; использование при обучении на примерах только словарных признаков, отобранных экспертами; задание весов различным терминам при формировании правил экспертами и реализация специальных инструментальных средств для упрощения работы экспертов-лингвистов по формированию правил.

References

1. *Kennedy A., D. Inkpen* (2006) Sentiment classification of movie reviews using contextual valence shifters. Computational Intelligence, Vol.22, No 2, pp. 110–125.
2. *Qiu L., Zhang W., Hu C., Zhao K.*. SELC: a self-supervised model for sentiment classification. Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09), New York, USA, 2009, pp. 929–936.
3. *Li S.* et al. Employing Personal/Impersonal Views in Supervised and Semi-supervised Sentiment Classification. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 414–423.
4. *Wan X.* Co-Training for Cross-Lingual Sentiment Classification. Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, 2009, pp. 235–243.
5. *Thomas M., Pang B., Lee L.* Get out the vote: Determining support or opposition from congressional floor-debate transcripts. Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 2006, pp. 327–335.
6. *Nakagawa T., Inui K., S. Kurohashi,* Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, 2010, pp. 786–794.
7. *He Y., Lin C., Alani H.* Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 2011, pp. 123–131.
8. *Vasilyev V. G.* 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, 2011, pp. 133–139.
9. *Marchand M., Shawe-Taylor J.* Learning with the set covering machine. Proc. 18th International Conf. on Machine Learning, 2001, pp. 345–352.
10. *Vasilyev V. G.* (2008) Complex technology of automatic text classification [Kompleksnaja tehnologija avtomaticheskoj klassifikacii tekstov]. Komp'iuternaia Lingvistika i Intellektual'nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii "Dialog 2006" [Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2008"]. Bekasovo, 2008, pp. 83–90.

Раздел III. Доклады, представленные участниками тестирования систем синтаксического анализа

В данном разделе публикуется итоговая статья организаторов тестирования систем синтаксического анализа и отдельные статьи участников тестирования. Полностью с комментирующими сообщениями участников можно ознакомиться на сайте конференции «Диалог»

ОЦЕНКА МЕТОДОВ АВТОМАТИЧЕСКОГО АНАЛИЗА ТЕКСТА 2011–2012: СИНТАКСИЧЕСКИЕ ПАРСЕРЫ РУССКОГО ЯЗЫКА[1]

Толдова С. Ю. (toldova@yandex.ru),
Соколова Е. Г. (minegot@rambler.ru)
РГГУ, Москва, Россия

Астафьева И. (astafir@gmail.com),
Гарейшина А. (a.r.gare@gmail.com),
Королева А. (tresh_miralissa@mail.ru),
Привознов Д. (dprivoznov@gmail.com),
Сидорова Е. (begushchaya.po.volnam@gmail.com),
Тупикина Л. (lyubov98@gmail.com)
МГУ им. М. В. Ломоносова, Москва, Россия

Ляшевская О. Н. (olesar@gmail.com)
НИУ ВШЭ, Москва, Россия

[1] Работа выполнена при частичной финансовой поддержке Программы фундаментальных исследований Президиума РАН «Корпусная лингвистика». Мы выражаем благодарность А. Бонч-Осмоловской, С. Ковалю, Ю. Гришиной, М. Ионову, А. Литягиной, Н. Меньшиковой, А. Семеновской, принимавшим вместе с нами участие в организации соревнований и экспертизе результатов. Отдельно мы выражаем благодарность организаторам конференции Диалог за помощь в проведении Форума и поддержку сайта (http://dialog-21.ru/Default.aspx?DN=605c0b84-a3bd-42d4-8225-b70760f65c1d&l=Russian). Также мы благодарим участников форума за сотрудничество.

Второй раунд форума «Оценка методов автоматического анализа текста» в 2011–2012 гг. был посвящен синтаксическим анализаторам русскоязычных текстов. В статье описываются принципы и процедура проведения дорожек форума, состав участников, тестовая коллекция и Золотой Стандарт, на основе которого осуществлялась оценка, принципы сопоставления ответов систем, сложные для оценки случаи, а также некоторые проблемные точки в работе синтаксических парсеров, которые выявила экспертиза результатов.

Ключевые слова: синтаксический анализ, автоматическая обработка текста, парсеры, русский язык, соревнования парсеров, оценка

1. Введение

В начале 2011 года был объявлен второй цикл форума «Оценка методов автоматического анализа текста». Темой форума стал автоматический синтаксический анализа русского языка. При организации форума 2011–2012 года использовался опыт Семинара по оценке методов информационного поиска РОМИП (ROMIP 2009) и форума 2010 года.

Целью Форума является создание независимой площадки, где представители научных, образовательных, коммерческих и т. п. организаций обсуждают состояние и перспективы развития алгоритмов и методов автоматической обработки текста (прежде всего, для русского языка), а также проводится экспертиза лингвистических компьютерных разработок. Независимые соревнования лингвистических систем проводятся в разных странах мира (ср. проекты CLEF, Morpho Challenge, AMALGAM, GRACE, EVALITA, PASSAGE, SEMEVAL, российский семинар оценки методов информационного поиска РОМИП и др.). В 2010 году состоялся первый цикл Форума, в котором приняло участие 15 команд разработчиков из Москвы, Санкт–Петербурга, Екатеринбурга, Украины, Беларуси и Великобритании. Форум–2010 был посвящен системам морфологического анализа русского языка (см. подробнее о принципах и результатах на сайте http://ru-eval.ru, а также Lyashevskaya et al. 2010). В рамках второго цикла оценивалось состояние лингвистических технологий в области автоматического синтаксического анализа. В России такое сравнение происходило впервые. На конференции «Диалог 2011» был проведен круглый стол с участием ведущих разработчиков синтаксических парсеров; осенью 2011 года состоялись дорожки форума. Как и в 2010 г., второй цикл форума имел и образовательную составляющую. В экспертной группе работали студенты, связывающие свое будущее с прикладной лингвистикой. Экспертиза результатов проводилась автоматически с последующей двойной ручной перепроверкой. Рейтинг ответов систем будет объявлен в мае 2012 г. на круглом столе конференции «Диалог».

Следует сразу отметить, что сравнение работы синтаксических анализаторов на порядок сложнее, чем оценка автоматического морфологического анализа. В области морфологии существует значительная зона пересечения: существуют общие представления о морфологической норме, отраженной в словарях и грамматиках, для большинства морфологических параметров есть устоявшаяся традиция «ярлыков», а значительная часть случаев, когда имеет место варьирование, поддается унификации простым переименованием тегов. В синтаксическом анализе могут быть использованы разные формализмы и принципы представления синтаксической структуры. В этом году сравнивались результаты работы систем, представленные в виде дерева зависимостей. Но и в этом случае результаты разбора сильно отличались друг от друга и в значительной степени зависели от того, каковы конечные задачи системы, в которую данный модуль встраивается. В связи с этим проведению самого соревнования предшествовал длительный этап подготовки, в том числе и обсуждения формата проведения форума в рамках конференции «Диалог 2011». Высокая активность как академических коллективов, так и промышленных разработчиков в процессе обсуждений показала, что данное направление автоматической обработки текста чрезвычайно востребовано на современном этапе. Помимо традиционной области применения результатов синтаксического анализа, такой как машинный перевод, данный модуль активно используется в системах автоматического анализа контента, например, в извлечении именованных сущностей или фактов из текста, при мониторинге блогов и новостей и др.

«Синтаксический» цикл проходил следующим образом. Участники форума получили специально отобранную и подготовленную коллекцию текстов, обработали их в своих системах и представили результат синтаксического анализа в некотором унифицированном формате. Правильность разбора оценивалась при сравнении с эталоном, размеченным вручную.

По результатам проведенного соревнования можно сказать, что, не смотря на различные трудности, с которыми организаторы столкнулись при проверке результатов, удалось выработать некоторый формат и принципы, позволяющие производить такое сравнение.

Форум не только позволил оценить работу синтаксических парсеров, но и дал целый ряд общезначимых в области синтаксического анализа результатов. Был получен корпус вручную размеченных и выверенных текстов, который можно использовать в научно–исследовательских целях (он представлен в свободном доступе на сайте testsynt.soiza.com). В подготовке и проведении дорожек форума 2011–2012 года и в формировании финального отчета активное участие принимали студенты Отделения теоретической и прикладной лингвистики филологического факультета МГУ им. М. В. Ломоносова, которые получили возможность «пощупать руками», как работают парсеры, увидеть, в чем их сильные и слабые стороны и т. д.

Как и при проведении форума 2010 года, в основу принципов проведения дорожек 2011–2012 гг. легло следующее положение: не бывает единственно правильного решения спорных вопросов и единственно правильного алгоритма синтаксического анализа. По возможности, ошибочными считались

только разборы, не мотивированные теоретическими или практическими установками авторов системы. Можно указать множество примеров того, как оптимальный выбор того или иного решения зависит от цели, для которой проводится анализ. Также существует целый ряд проблемных случаев, не имеющих единственного решения. При сравнении работы разных парсеров был уточнен список проблемных зон в области синтаксического анализа, а также множество возможных подходов к их обработке.

Таким образом, в процессе проведения форума удалось получить некоторую оценку состояния автоматического синтаксического анализа русского языка, выявить проблемные и дискуссионные места синтаксического анализа, в которых при разных подходах принимаются принципиально разные решения, оценить варьирование в базовых подходах к типизации синтаксической реальности. Также результаты форума показали, что в области автоматического синтаксического анализа русского языка разработчикам удалось достичь достаточно высокого уровня.

2. Подходы и проблемы, связанные с оценкой автоматического синтаксического анализа

Предварительная оценка состояния автоматического синтаксического анализа для русского языка показала, что большинство систем используют формализм зависимостей. Таким образом, при проведении конкурса рассматривались результаты, представленные в виде деревьев зависимостей, независимо от тех формализмов, которые использовали разработчики в своих системах.

При организации синтаксического цикла мы опирались на мировой опыт проведения соревнований подобного типа, некоторые из которых упомянуты во Введении, в частности, на опыт проведения аналогичной оценки систем для итальянского языка EVALITA. Для дорожки по деревьям зависимостей участники получают на вход корпус текстов, разбитых на предложения и токены. Задача заключается в том, чтобы для каждой словоформы в предложении указать ее синтаксическую вершину, а также тип синтаксической связи.

Как правило, при проведении соревнований множеством эталонного набора типов связей (имен связей и набор устанавливаемых синтаксических отношений), используемые в оценке работы систем, служат данные уже готовых синтаксически размеченных корпусов. Тем более, многие разработчики используют эти наборы при создании систем, особенно если система строится на машинном обучении. Так, например, для итальянского языка используется Turin University Treebank (TUT), размеченный в обоих формализма (и в терминах непосредственных составляющих, и в терминах деревьев зависимостей)[2].

[2] Широко известны и активно используются в обучении анализаторов и оценке их работы для английского языка Penn Treebank, размеченный по непосредственным составляющим, The Prague Dependency Treebank для чешского языка, основанный на деревьях зависимостей.

Также предложения из таких трибанков часто служат тестовым корпусом, что позволяет обеспечить процедуру автоматической проверки.

Анализ пробного разбора 100 предложений, представленного разработчиками–потенциальными участниками Форума 2011–2012, показал, что в России системы синтаксического анализа развивались автономно, без использования какого бы то ни было корпуса в качестве эталона. В результате, расхождения между системами по составу тегов и по принципам установления связей оказались настолько значительными, что в целом ряде вопросов не удалось предложить единого решения для представления выходных данных. Было принято решение о том, что на данном этапе оцениваться должно только правильное определение системами синтаксически связанных пар словоформ и установление «главного» элемента в паре. При этом при оценке не должны оцениваться теоретические расхождения в трактовке тех или иных синтаксических явлений.

3. Форум 2011–2012: синтаксические парсеры

3.1. Дорожки

На форуме 2011–2012 по синтаксическому анализу текстов оценивание алгоритмов систем–участников прошло независимо по следующим отдельным дисциплинам (дорожкам):

- «общая»; в этой дорожке рассматривались различные типы текстов и синтаксический разбор всех представленных в них предложений;
- «новостная»; задача этой дорожки состояла в синтаксическом разборе предложений узкой тематики, а именно — новостного блока.

В процессе подготовки форума также затрагивались вопросы о дальнейшей разработке дополнительных дорожек — по разбору сложных предложений целиком vs. отдельному разбору простых предложений в составе сложного, выделению проективных vs. непроективных предложений и др. Однако проведение таких дорожек сильно бы усложнило и без того достаточно трудоемкую процедуру проверки.

3.2. Участники

На конкурс были поданы заявки от 11 различных групп разработчиков из Москвы, Санкт–Петербурга, Нижнего Новгорода (Россия), Донецка (Украина). Одна из этих групп участвовала в проекте вне конкурса, поэтому её результаты не включались в общее соревнование. Конечные результаты, и по основной, и по новостной дорожкам, были получены от 8 из 10 участников форума: SynAutom, DictaScope Syntax, SemSin, ЭТАП–3, синтактико–семантический

анализатор русского языка группы SemanticAnalyzer Group, проект AotSoft, ABBYY Syntactic and Semantic Parser (ASSP), Парсер грамматики связей. Среди них системы, использующие различные методы синтаксического разбора: грамматику зависимостей, грамматику составляющих, грамматику связей (Link grammar parser). Один из восьми разработчиков впоследствии был вынужден отозвать своё участие в конкурсе из–за проблем с конвертированием данных. Таким образом, в окончательной оценке участвовало 7 различных систем обработки текстов.

3.3. Тестовая коллекция и задания

Для соревнования была подготовлена общая коллекция неразмеченных текстов. В коллекцию для «основной» дорожки вошли тексты разных жанров, включая художественную литературу, публицистику, а также 5 % текстов из социальных сетей. В коллекции были представлены как отдельные предложения (200 тыс. словоупотреблений из Национального корпуса русского языка, предоставленные для свободного скачивания), так и фрагменты связанных текстов. В новостную коллекцию вошли фрагменты текстов из новостной коллекции семинара РОМИП. В эту коллекцию попали последовательности из трех предложений, выбранные случайным образом. Все тексты были заранее разбиты на предложения и токены и проиндексированы.

Участники конкурса должны были приписать каждому токену номер его вершины. При проверке не оценивалась правильность разбора всего предложения, оценивалась правильность приписывания вершины зависимой словоформе. Сравнение результатов по всем дорожкам проводилось на основе выборочной проверки ответов систем–участников. Для этого был подготовлен «Золотой Стандарт» — множество случайно выбранных предложений из Основной коллекции, объемом около 800 предложений (500 для основной коллекции и 300 для новостной). В ходе экспертизы ответы систем сравнивались с произведенной экспертами ручной разметкой Золотого Стандарта, см. п. 3.7–3.8.

3.4. Соглашения по унификации входного формата

Для унификации результатов, получаемых от разных систем, был разработан специальный входной формат представления текстовой коллекции. Исходный корпус предоставлялся участникам в двух форматах: исходный текст без разметки и xml–формат с разбивкой на предложения и токены. Были приняты некоторые соглашения относительно правил токенизации. Отдельными токенами считались словоформы, входящие в одну сложную единицу, например, в сложный союз или предлог. На отдельные токены также разбивались слова с дефисом, за исключением некоторых заданных списком слов, а также местоимений с частицей *–то*, наречий с *по–* и т. п., отдельными токенами считались знаки препинания.

Предварительная токенизация и нумерация токенов нужна была для того, чтобы минимизировать долю ручной проверки. Благодаря такой унификации можно было автоматически определять фрагменты, в которых разметка участников совпадает с Золотым Стандартом, что минимизировало долю ручной проверки. В первую очередь «вручную» просматривались места несовпадений. Многие участники игнорировали нумерацию в процессе работы собственного анализатора, но потом приводили ID токенов в соответствии с ID токенов в тестовой коллекции.

3.5. Соглашения об унификации выходного формата

Результат работы систем должен был быть представлен также в специальном формате. В выходном файле нумерация предложений и токенов должна была соответствовать нумерации в тестовом корпусе. Участники должны были указать для каждой словоформы номер «хозяина» (главного слова в словосочетании) и тип связи (указывался тип синтаксической связи, принятый у разработчика), также указывалась морфологическая информация: лемма и набор морфологических характеристик. Тип связи и морфологическая информация указывалась на усмотрение разработчиков и нужна была для облегчения ручной проверки, чтобы эксперту было легче понять, в чем причина расхождения ответа системы с Золотым Стандартом.

3.6. Соглашения по унификации направлений связей

Подготовительный этап потребовал определенных решений, направленных на унификацию структуры синтаксических отношений в ответах, ожидаемых от парсеров. Существует достаточно много ситуаций, когда системы по–разному решают вопрос о направлении синтаксической зависимости между двумя словоформами, находящимися в отношении синтаксической связи (подробнее о расхождениях см. п. 4). Эти случаи обусловлены не ошибками при анализе, а принципиальными решениями при создании конкретных систем. В таких случаях расхождения системы с эталоном не «штрафовалось». Однако для того, чтобы не пришлось просматривать каждый подобный случай вручную, некоторые системы, по крайней мере, в части случаев согласились изменить направления связей там, где это было возможно сделать автоматически. Это касалось следующих типов связей:

1) предлог — существительное;
2) вспомогательный глагол — смысловой глагол;
3) связи в сочинительных конструкциях.

3.7. Подготовка Золотого Стандарта

Разметка Золотого Стандарта, предшествовавшая экспертизе результатов, проводилась вручную с помощью инструмента для разметки, подготовленного

М. Ионовым. Каждое предложение первоначально размечалось двумя экспертами, после чего места расхождений обсуждались. На основании обсуждений принималось единое решение. Далее окончательный вариант проверялся третьим экспертом. Такая разметка позволяла достичь нескольких целей. Во–первых, это позволило автоматизировать процедуру разметки. Во–вторых, организаторы хотели по возможности избежать влияния результатов, предоставленных системой, на интуицию экспертов, и пропусков ошибок по невнимательности. В–третьих, разметка Стандарта должна была сформировать у экспертов представление о том, какие сложные случаи их ожидают, выработать критерии для оценки расхождений.

При разметке аннотаторы пользовались специальной инструкцией, обеспечивающей «устойчивость» аннотации, т. е. согласованность в принятии решений разными аннотаторами в одинаковых ситуациях.

3.8. Принципы разметки Золотого Стандарта

Для разметки Золотого Стандарта требовалась такая инструкция, которая бы обеспечила не столько теоретическую (абстрактную) «правильность» разметки, сколько единообразие разметки разными аннотаторами, четкую обоснованность принимаемых при разметке решений.

Мы основывались на принципах и средствах синтаксической разметки, сформулированных в (Sokolova 2011; ср. также Hovy and Lavid 2010) и опробованных на занятиях по синтаксической разметке текстов студентов 4–го курса РГГУ в течение нескольких лет. Одним из важных принципов, который лег в основу принимаемых решений, является принцип «естественности» разметки: разметка должна соответствовать правильной семантической интерпретации предложения (в инструкции он формулируется следующим образом: «Синтаксическая структура языкового произведения осмысленна и единственна»). Из возможных вариантов отбирались наиболее простые и понятные решения, которые максимально согласовались с интуицией разметчика.

Форма структуры — дерево зависимостей, узлами которого являются словоформы (а не наборы морфологических интерпретаций словоформ). При этом может сохраняться некоторая неоднозначность ее морфологической и семантической интерпретации, не противоречащая структуре дерева, например, в предложении *Свидания разрешить не могу* словоформа *свидания* зависит от словоформы *разрешить* и имеет синтаксическую функцию “obj”. В структуру может входить любая из двух ее морфологических интерпретаций: — *свидание* — ед. ч., род. п. в контексте отрицания (т. е. «не разрешаю конкретное свидание»); — мн. ч., вин. п. (т. е. «вообще никакие свидания не разрешаются»). В отличие от «классической» Грамматики Зависимостей интерпретация отношений заменена на Синтаксические Функции словоформ — формализм ГЗиСФ, при котором синтаксическая функция приписывается всем словоформам — и подчиненным, и вершинам. Принцип единственности структуры заимствован из опыта разметки целых текстов студентами, при которой предложение рассматривается внутри конкретного текста. Мы сохранили этот принцип для разметки отдельных предложений

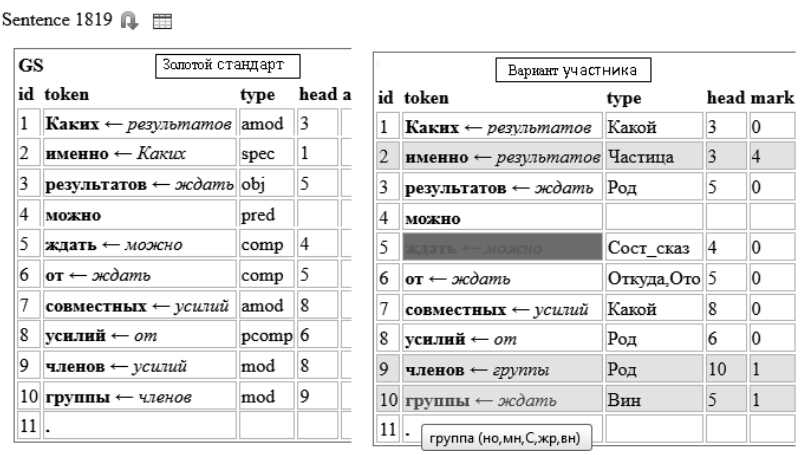
в Золотом Стандарте. При этом редкие (относительно Золотого Стандарта) варианты синтаксической интерпретации предложения не считаются ошибкой.

3.9. Экспертиза ответов систем

Процедура экспертизы ответов синтаксических анализаторов предусматривала сравнение номеров вершин, указанных системами для каждой словоформы, с ее номером в Золотом Стандарте. Совпадение номеров автоматически получало оценку 0. Случаи расхождений просматривались экспертами, которые должны были оценить их по следующей шкале:

1 — ошибка системы;
2 — ошибка ЗС;
3 — допустимое расхождение (расхождения объясняются расхождением в теоретических решениях системы и ЗС);
4 — допустимое расхождение (случай допустимой омонимии);
5 — ответ системы совпадает с ЗС, но оба неправы;
6 — для данного токена «хозяин» не указан, а должен быть указан;
7 — для данного токена «хозяин» не указан и может быть не указан;
8 — затрудняюсь определить (эксперт не может принять однозначное решение);
9 — другое.

Фрагмент проверочной таблицы указан на рисунке 1[3].


Sentence 1819

GS (Золотой стандарт)

id	token	type	head	a
1	**Каких** ← *результатов*	amod	3	
2	**именно** ← *Каких*	spec	1	
3	**результатов** ← *ждать*	obj	5	
4	**можно**	pred		
5	**ждать** ← *можно*	comp	4	
6	**от** ← *ждать*	comp	5	
7	**совместных** ← *усилий*	amod	8	
8	**усилий** ← *от*	pcomp	6	
9	**членов** ← *усилий*	mod	8	
10	**группы** ← *членов*	mod	9	
11	.			

Вариант участника

id	token	type	head	mark
1	**Каких** ← *результатов*	Какой	3	0
2	**именно** ← *результатов*	Частица	3	4
3	**результатов** ← *ждать*	Род	5	0
4	**можно**			
5	**ждать** ← *можно*	Сост_сказ	4	0
6	**от** ← *ждать*	Откуда,Ото	5	0
7	**совместных** ← *усилий*	Какой	8	0
8	**усилий** ← *от*	Род	6	0
9	**членов** ← *группы*	Род	10	1
10	**группы** ← *ждать*	Вин	5	1
11	.	группа (но,мн,С,жр,вн)		

Рис. 1. Пример разметки Золотого Стандарта и ответа одной из систем. В графе “mark” указана оценка за решение

[3] Организаторы выражают благодарность Горшкову Д. В. за компьютерную поддержку в проведении конкурса, в частности за разработку базы данных и визуализации деревьев для обеспечения автоматического и ручного этапа проверки, а также повторной перепроверки.

Сравнение ответов систем с Золотым Стандартом позволило выделить наиболее распространенные отклонения от разборов, признанных эталонными (см. п. 4.1).

Проверка также показала, что не всегда удается оценить, в какой степени тот результат, который представлен в ответе системы, определяется принципиальными решениями, принятыми в системе, проблемами «пересчета» направлений связи в соответствии с Золотым Стандартом или же ошибкой в разборе. К сожалению, таких случаев оказалось значительное количество. Они потребовали дополнительной выверки результатов. Значительную помощь в улучшении системы оценки оказали комментарии разработчиков, присланные ими после того, как они получили доступ к промежуточным оценкам. Однако даже при дополнительном пересмотре не удалось избежать ситуаций, когда «штраф» системе приписан ошибочно.

В следующем разделе остановимся более подробно на отдельных вопросах выработки ряда решений при организации Форума 2011–2012, а также на сложных моментах, с которыми нам пришлось столкнуться.

4. Трудные случаи и расхождения

4.1. Допустимая вариативность разборов

Расхождения между системами по составу тегов и по принципам установления связей оказались настолько значительными, что в целом ряде вопросов не удалось предложить единого решения для представления выходных данных.

Во–первых, разные системы не только используют разные названия для одних и тех же синтаксических отношений, но существуют значительные расхождения в самой классификации типов связи. Так, в одних системах разграничение типов связей опирается на морфологическую разметку, в других, наоборот, учитывается самая общая синтаксическая функция словоформы. Например, в одних системах отдельно выделяется тип связи “card” для связи числительного с существительным (ср. *тысячи* ← *педагогов* (card)), в других этот случай относится к общему случаю несогласованного определения. В силу этого обстоятельства решено было при сравнении результатов не учитывать имена связей.

Во–вторых, помимо конструкций, не вызывающих вопросов и размечаемых всеми одинаково (согласованное определение), существуют конструкции, относительно которых не существует единого теоретического решения. В частности, в целом ряде конструкций невозможно однозначно установить, какой из синтаксически связанных элементов является главным, а какой зависимым (подробно о таких конструкциях см., например, Iomdin 1990, Gladkij 1973): это случаи, когда либо разные критерии выделения вершин дают разные результаты (см., например, Testelets 2001), либо ни один критерий не применим. Примером может служить сочинение: при наличии союза между сочиненными

элементами количество различных разборов становится немалым, потому что этому союзу можно приписать несколько разных вершин (а также считать вершиной сам союз). Однако до тех пор, пока все сочиненные члены с союзом или союзами соединяются в одну группу, нет причин считать такой разбор ошибкой. Несколько вариантов разбора допустимы также в случае становления связи между клаузами в сложноподчиненных предложениях. В ряде систем клаузы соединяются между собой через глаголы, в других — через подчинительные союзы.

В третьих, вариативность в разборах обусловлена разными практическими задачами, решаемыми системами. Так, например, в соответствии с критериями выделения вершин главным в словосочетании ‘вспомогательный глагол + смысловой’, как в *станет писать*, является вспомогательный глагол, однако многие системы последовательно устанавливают направление связи в данном случае ‘вспомогательный глагол ← смысловой глагол’.

В результате анализа разборов Е. Г. Соколовой была составлена таблица возможных расхождений по отдельным типам связей (см. http://testsynt.soiza.com/files/var-synt.htm).

В дальнейшем целесообразно добиваться того, чтобы ответы систем одинаково представляли наиболее частотные случаи, в которых сейчас наблюдаются расхождения: неодносложные союзы и предлоги, сложные слова с дефисным написанием; связь между однородными членами, между главной и подчиненной клаузой, между сочиненными клаузами (включая интерпретацию союзов), союз в начале главной клаузы; глагол-связку с инфинитивами, именами, прилагательными, причастиями; группы с количественными и порядковыми числительными (включая предложные и с модификаторами типа *более, минимум*); связь подлежащего с именным сказуемым; связь в группах вида ‘прилагательное + прилагательное + существительное’ и нек. др.

4.2. Анализ ответов систем: проблемные точки

В целом, приятным итогом анализа ответов стал вывод, что в пределах простого предложения/клаузы нет «больных мест», общих для всех участников. Среди частных проблем можно назвать свободно присоединяемые предложные зависимые (или те, что отсутствуют в актантном словаре или не выучены системой). Если в предложении находится несколько потенциальных хозяев, то системы выбирают либо линейно предшествующее существительное, либо вершинный глагол, либо ближайший финитный глагол в дереве, однако не все такие варианты будут семантически оправданы, ср. допустимые (1А–В), (2А–Б) и недопустимые (1Г), (2В):

(1) *Компания Google продолжает укреплять свои позиции на рынке приложений для совместной работы.*
А. ^{ок} *позиции → на рынке*

Б. [ок] *укреплять → на рынке*
В. [ок] *приложений → для совместной работы*
Г. * *укреплять → для совместной работы.*

(2) *… что может добиться своей цели лишь при одном условии…*
А. [ок] *добиться → при условии*
Б. [ок] *может → при условии*
*В. *цели → при условии*

Большинство систем не смогло справиться с примером, в котором присутствуют три однородных определения вида X, Y и Z к существительному:

(3) *В качестве пилотных субъектов РФ признаны Челябинская, Томская и Архангельская области.*

Системы могут ошибочно считать, что первые определения зависят от РФ, или не найти связи с несогласованным по числу существительным.

Многие системы ошибаются при обработе идиоматических конструкций «малого синтаксиса», если срабатывают альтернативные характерные для русского языка шаблоны, ср. неверно приписанную атрибутивную связь в паре *обучение → такое* (4):

(4) *Что такое обучение?*

В сложных предложениях, безусловно, ошибок больше. Часто наблюдаются проблемы с нахождением вершины в предшествующей клаузе. Например, в (5) хозяевами вершины клаузы *чтобы… двигалась…* называются *возьмем*, *образуем*, но не элементы в составе деепричастного оборота. Аналогично, могут оставаться незамеченными вершины–существительные или связки типа есть.

(5) *Если мы возьмем какую–то замкнутую фигуру и образуем твердое тело, вращая эту фигуру в пространстве так, чтобы каждая точка двигалась перпендикулярно к плоскости фигуры…*

Наконец, во многих случаях наблюдается ложное срабатывание систем, когда дистантно расположенный зависимый выхватывается через границу клаузы, а также ненахождение связей для несловарных слов (например, ОС, Intel и др.)

5. Заключение. Итоги форума

На наш взгляд, несмотря на большую вариативность в теоретических подходах, практических решениях, качестве работы систем, несмотря на то, что

не удалось конвертировать ответы систем в единый формат, который можно было бы автоматически сопоставить с Золотым Стандартом, проведение форума синтаксических парсеров дало много полезных результатов:

- был создан вручную размеченный эталон объемом в 800 предложений, а также передана в общее пользование инструкция, эксплицитно поясняющая те или иные решения
- для систем, представивших результаты, был создан реестр расхождений, который может быть обобщен до реестра допустимых общетеоретических решений и таблицы их «эквивалентности»
- была осознана необходимость «публичности» эталонного общезначимого трибанка с параллельной разметкой разными системами, аккумулирующая множества тегов и принципов разбора; особенно ценен такой ресурс для разработчиков, чьи знания о синтаксисе не выходят за рамки школьной программы, а также для развития систем, «варящихся в собственном соку»
- разработчики систем получили открытый доступ к своим промежуточным оценкам; по сравнению с форумом 2010 года, удалось добиться большего взаимодействия организаторов и разработчиков при подготовке дорожек и обсуждении результатов
- главный итог: до начала объявления соревнований трудно было оценить общую ситуацию с состоянием автоматического синтаксического анализа в России: какие системы представлены, какие формализмы используются, какие принципы установления синтаксической зависимости между единицами предложения положены в основу, каково множество синтаксических отношений, которые система различает. Проведенное соревнование позволило, в определенной степени, такую картину составить.

Перспективы продолжения форума нам видятся в дальнейшей автоматизации экспертизы и еще большей ее открытости; в том, чтобы повысить гибкость оценок с учетом комментариев по принципиальным решениям, высказанных разработчиками; в содержательном ключе хотелось бы большее внимание уделить типам синтаксических связей и обработке сложных предложений. А у разработчиков, в свою очередь, будет возможность улучшить результаты с учетом накопленного соревновательного опыта.

References

1. *Gladkij A. V.* (1973), Formal'nye grammatiki i jazyki [Formal Grammars and Languages], Moscow, Nauka.
2. *Hovy E., Lavid Ju.* (2010), Towards a 'science' of corpus annotation: a new methodological challenge for corpus linguistics, International journal of translation, Vol. 22, no. 1, pp. 1–25.
3. *Iomdin L. L.* (1990), Avtomaticheskaja obrabotka teksta na estestvennom jazyke: model' soglasovanija [Natural Language Processing: a Model of Agreement], Moscow, Nauka.

4. *Lyashevskaya O., Astaf'eva I., Bonch–Osmolovskaya A., Garejshina A., Grishina Ju., D'yachkov V., Ionov M., Koroleva A., Kudrinsky M., Lityagina A., Luchina E., Sidorova E., Toldova S., Savchuk S., Koval' S.* (2010), Ocenka metodov avtomaticheskogo analiza teksta: morfologicheskije parsery russkogo jazyka [NLP evaluation: Russian morphological parsers], in Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue'2010, Vol. 9 (16), Moscow, pp. 318–326.
5. *ROMIP* (2009): Rossijskij seminar po ocenke metodov informacionnogo poiska. Trudy ROMIP 2009, Petrozavodsk, 16 sentjabrja 2009 [Russian Information Retrieval Evaluation Seminar. Proceedings of ROMIP 2009, Petrozavodsk, September 16, 2009]. Saint–Petersburg, NU CSI.
6. *Sokolova E. G.* (2011), Syntactic annotation in terms of dependency grammar and syntactic functions [Sintaksicheskaja razmetka v terminax grammatiki zavisimostej i sintaksicheskix funkcij], Moscow, RGGU, available at: http://elib.lib.rsuh.ru/elib/000003603.pdf
7. *Testelets Ja. G.* (2001), Vvedenie v obshchij sintaksis [Introduction to general syntax], Moscow, RGGU.

SYNTACTIC AND SEMANTIC PARSER BASED ON ABBYY COMPRENO LINGUISTIC TECHNOLOGIES

Anisimovich K. V. (Konstantin_An@abbyy.com),
Druzhkin K. Ju. (Konstantin_D@abbyy.com),
Minlos F. R. (f.minlos@gmail.com),
Petrova M. A. (Maria_P@abbyy.com),
Selegey V. P. (Vladimir_S@abbyy.com),
Zuev K. A. (Konstantin_Z@abbyy.com)

ABBYY, Moscow, Russia

The paper presents ABBYY Syntactic and Semantic Parser that was a participant of the Dialog 2012 Syntactic Parsers Testing Forum. We will refer to the parser technology (both parsing algorithms and linguistic model) as Compreno technology. We do not touch on any evaluation issues, as they are tackled by the Forum panel. Instead, the paper makes public some underlying principles of the parser. What we want to communicate directly concerning the testing are the features of the project which are both relevant to the comparison of our results with the "gold standard" adopted by the panel and, at the same time, important for the whole architecture of our technology.

Key words: syntax, semantics, natural language processing, parser

Introduction

Essentially, what a parser strives to extract from the sentence is *who did what to whom* (*when* and *where*). The question is what level of representation is aimed at. In Compreno project, the ultimate goal is to achieve not only the syntactic disambiguation, but the semantic one as well. Semantic and syntactic representations are viewed rather as two facets of the same structure (much as in the mainstream G&B / P&P approach), than as two distinct types of structure (as, for instance, in LFG). Another (interrelated) feature of the Compreno parsing technology is that syntactic and semantic disambiguation are processed in parallel from the very start (in contrast to the architecture more usual for the NLP systems — the semantic analysis follows the syntactic one). This Compreno peculiarity makes it difficult to adapt analysis structures for the surface syntax testing requirements.

The objective of immediate semantic interpretation for a syntactic structure determine some features of the syntactic structures in question. The most evident one is the heavy use of null elements. The model case of null element is phonetically null subject of non-finite clause. For example, in equi-predicates and raising

predicates, the null subject of infinitive is coreferential with an argument of the matrix predicate.

Another huge class of mismatches also comes from different perspectives, taken by the Compreno project and the Golden Standard (GS). Within the purely syntactic framework of the GS, functional words (such as prepositions, conjunctions, and complementizers) are treated as heads, while in Compreno syntax, they are dependent nodes. The same goes for a little more complicated **более чем**-construction (*в течение более чем сорока лет*; *подавать документы в более чем пять вузов*). The GS picks up *более* as the head of the construction (a natural decision, within the syntacticocentric approach), for Compreno, the semantic link to the noun is more important (for example, we immediately obtain the collocation *подавать документы в вуз(ы) 'submit documents to institutes'*, without any intervening nodes).

A final example of discrepancy to present here is clause attachment in examples like *Не вызывает сомнения, что…* 'No doubt…'. Compreno parser attaches the *что*-clause to the noun licensing it; thus we can count statistics concerning the frequency of *что*-clauses with specific nouns.

The task of the normalization of Compreno syntactic structures for the parser competition was all but trivial, taking into account striking difference between the two syntactic representation (when two or more discrepancies occurred in the same clause, the situation could became even more complicated).

The Compreno Linguistic Model

1. Basic syntax

1.1. Dependency links and constituent structure

The syntactic structure of a sentence is represented as a syntactic tree, augmented with non-tree links. Tree links encode syntactic dominance; non-tree links capture conjunction, anaphora, distant agreement, and other non-local dependencies between nodes.

The syntactic trees modeled within the Compreno framework may be seen either as projective dependency trees or, alternatively, as constituent trees, where every non-terminal node has one terminal child (its **lexical head**, or **core**) and zero or more non-terminal children.

Projective dependency trees are such that their subtrees correspond to contiguous chunks of text (which means that if A and C are descendants of X, and B is linearly between A and C, then B is also a descendant of X). This property allows us to speak of subtrees and constituents interchangeably[1].

[1] Projective dependency trees are obviously isomorphic to constituent structures with lexical heads: "words" correspond to "lexical cores", "subtrees" to "constituents", "dependency links" to the relations between the (cores of) constituents and the (cores of) their child constituents.

Children fill certain syntactic slots within the parent (e. g. articles fill the “$Article” slot within noun phrases; the link between a verb and its subject is labeled "$Subject” and so on). Most syntactic links also get a semantic interpretation, with the exception of function words (such as auxiliary verbs, conjunctions, or prepositions), and displaced elements (see below).

We treat conjuncts as sisters, i. e. daughters of a common parent.

1.2. Linear order

As in HPSG, the description of possible linear orders is kept separate from the description of possible constituent structures. Suppose, `{Slot _ 1 ... Slot _ N}` is a set of syntactic slots, available inside a noun phrase. E.g. $Article is in this set, and also $Modifier_Attributive, $OfPostmodifier and many others. A **linear order template** would look like this:

(9) `Slot _ 1 [Slot _ 2 Slot _ 3] Core Slot _ 4 (Slot _ 5 | Slot _ 6) Slot _ 2`

Not all slots must be filled, but if they are filled, then the fillers must be in this order with respect to the core and to each other. Square brackets mean that any order is allowed. Vertical bar means that only one of the alternatives is allowed.

A linear order description is a set of variants. A variant is a pair of:

- a grammar expression that is matched against the grammar value of a constituent,
- a linear order template.

Variants are needed because linear orders may vary (e. g. in declarative and interrogative sentences).

A number of inferences can be made from such descriptions. For example, we may know that some slot is never allowed before the core in noun phrases. Such inferences are automatically extracted, stored away and later used by the parser.

1.3. Morphological and syntactic categories

A morphological paradigm is a multi-dimensional table, where dimensions are the morphological categories (= attributes), and columns or rows are morphological grammemes (= values).

Ideally, all cells in the table must be filled with single word-forms. But how should we deal with analytical verb forms (e. g. futures or passives that require auxiliary verbs)? If we put them in the table, we leave the realm of morphology proper. Auxiliary verbs fit into the internal structure of verb phrases (in syntax), not into the internal structure of verbs (in morphology).

This approach can be generalized to other grammatical elements as well. If auxiliary verbs are part of verbal paradigm, then prepositions are part of nominal

paradigm. So in addition to the morphological category of Case with grammemes <Nominative | Genitive | Dative | ...>, we now have the syntactic category of ExtendedCase with grammemes <ECabout | ECafter | ... | ECwithout>.

Finally, for every syntactic slot $Slot we can create a binary category SlotCategory with grammemes <SlotFilled | SlotNotFilled>. For instance, noun phrases that contain an "of"-postmodifier ("[the father [of John]]") get the <OfPostModifier> grammeme.

The move from purely morphological inflectional paradigms to morphosyntactic paradigms including analytical forms (of, e. g., tense) is a traditional one. What is unusual in the Compreno approach is the inclusion of the absolutely asymmetrical oppositions in the paradigm. E.g., the absence of a modifier is not likely to be perceived as a zero instantiation of some binary opposition. But in Compreno project, the grammeme is a universal means to refer to any syntactic configuration (see also the next chapter).

1.4. Syntactic levels and syntactic forms

At the core of the Compreno framework lies the notion of **syntactic paradigm**. The notion may at first sound odd, but it is a natural extension of the traditional notion of paradigms in morphology. Moreover, this move is not a new one. The term *syntactic paradigm* was used by Kenneth L. Pike [4][2]. The theoretical perspective of notions morphological paradigm vs. syntactic paradigm is discussed in detail in [6].

The set of categories and available grammemes for a certain part of speech is composed of multiple sources. First, there are **morphological categories** that come from the morphological dictionary (e. g. case or gender of adjectives). Second, there are **classifying categories for lexemes**; their grammemes are assigned manually to some lexemes that have non-trivial syntactic properties (e. g. Russian numerals from 2 to 4 are marked <SmallNumeral>). They can be viewed as extensions to morphology. Third, there are **classifying categories for classes**. Fourth, there are **syntactic categories**, to be discussed here. Finally, there are special categories, such as Capitalization; their values are supplied by the processing engine.

A pair of a lexeme and a lexical class uniquely selects **a syntactic paradigm**, which defines a full range of syntactic possibilities allowed.

A paradigm is a set of **syntactic levels**, each of which defines some aspect of syntactic structure. Logically, a paradigm is a conjunction of levels: the constituent has to match all of them. Some levels are universal (i. e. defined for parts of speech, e. g. for all noun phrases), some lexicalized (i. e. defined for branches of semantic hierarchy).

A syntactic level is a set of **syntactic forms**, each of which defines a specific syntactic configuration. Logically, a level is a disjunction of forms: the constituent has to match at least one of them.

---

[2] The founder of tagmemic grammar; he also coined the term grammeme, heavily used in Russian linguistic, including the Compreno project, but almost uknown in the Western tradition, see [12].

Levels are associated with categories of syntax; forms are associated with specific grammemes. If a constituent matches a form, it gets a corresponding grammeme.

A syntactic form may specify, among other things:

- a grammar expression that is matched against the grammar value of a constituent,
- zero or more surface slots that must be filled,
- for each surface slot, a set of semantic slots, available as its semantic interpretation.

There are syntactic categories that are parallel to morphological categories. For example, verbs have morphological Tense. Bun in analytical forms, the Tense is morphologically expressed on the auxiliary verb. So we have a syntactic category SyntacticTense. When the *main* verb is finite, its SyntacticTense follows its Tense; but when the *auxiliary* verb is finite, the main verb copies its value of SyntacticTense by agreement.

1.5. Underspecification

A word-form (or a constituent) can be *underspecified* with respect to some category. For example, plural forms of Russian adjectives are underspecified for Gender.

In many cases, morphological ambiguity is resolved by syntactic context; but sometimes it must remain. For example, some Russian nouns have identical forms for Genitive and Accusative, and in some constructions both Genitive and Accusative are allowed.

If a morphological ambiguity cannot be resolved, we say that the grammar value is underspecified in some category.

Underspecification is intimately connected to three-valued logic. (Indeed, the question "Is this word-form Genitive?" now has three possible answers: "Yes", "No" and "Maybe"). Also, it naturally lends itself to use in unification algorithms (more of which later).

1.6. Grammar expressions

The name of a grammeme can be viewed as a predicate, and such predicates can be combined by standard means of predicate logic: conjunction, disjunction, negation and bracketing. So we get **grammar expressions**, as the following[3]:

(3) `~Present | Imperfective`

(4) `~(Present, ~Imperfective)`

Grammemes in a grammatical category form a set; so negation means complementation in that set. Suppose a category `A` with grammemes `{a1, a2, a3}`. The expression `<~a1>` is equivalent to `<a2 | a3>`.

Grammar expressions are used in many places. Most importantly, they restrict the set of possible parent-child combinations in syntactic trees.

[3] Those are two ways to express logical implication.

1) Syntactic slots (or, alternatively, links) are special objects with their properties. The major property of a slot is "government", which is a grammar expression describing the allowed form for the child node.

2) Syntactic slots are connected to "syntactic forms" of the parent constituent. The concept of "syntactic forms" will be explained later. What is important now is that syntactic forms have grammar expressions describing the parent node. For example, comparisons (*"than X"*) can be attached to adjectives, only when adjectives have comparative degree (*"bigger than X"*, but not *"big | biggest than X"*).

When a grammar value of a would-be child is matched against a grammar expression of the slot, the expression can return "false" or "true". "True" means "possibly true", because the grammar value can be underspecified.

Suppose we check if an indeclinable noun can be a dative object. The government of the slot demands `<Dative>`, but the noun has `<Nominative|Genitive|Dative|...>`. The check returns "possibly true", but the contradictory grammemes remain in the grammar value of the noun. But when, at some stage of parsing, we commit to having this link in the tree, the contradictory grammemes `<Nominative|Genitive|...>` are filtered from the grammar value, and only `<Dative>` remains.

1.7. Agreement

An agreement rule is a set of variants. An agreement variant is a triple of:

1. a grammar expression, that is matched against the grammar value of the first node;
2. a grammar expression, that is matched against the grammar value of the second node;
3. a list of agreement categories, in which the two nodes must have the same grammar value.

Logically, an agreement rule is a disjunction of its variants, and a variant is a conjuction of its three parts.

Agreement is checked between two nodes, connected in some way. The rules of agreement are auxiliary to the rules that create connections. Depending on the type of the main rule, the "first and second nodes in agreement" can be parent and child, or left conjunct and right conjunct, or noun phrase and anaphoric pronoun that refers to it.

2. Null elements

2.1. Motivation for null elements

The use of null elements in linguistic frameworks is contentious. On the one hand, null elements simplify syntactic rules, and make surface structures closer to their semantic interpretations. On the other hand, null elements remain a theoretical

construct, not directly supported by the observable data, and, more importantly, they come with a high computational cost.

From the algorithmic point of view, null elements fall into two groups. Null elements that have no "real" descendants are not important in the early stages of parsing. The decision about their existence can be postponed until a syntactic tree has been built. Such unproblematic cases will be discussed in the next section. Null elements that can have "real" descendants make more trouble, leading to an undesirable proliferation of hypotheses at early stages of parsing. So it is worthwhile to consider the reasons for having null cores.

The substantivation of adjectives, as in (12), and coordinate ellipsis, as in (13), can be easily analyzed both ways:

(12) *The politicians and **the rich** did not bother.*

(13) *It improves the flow of traffic not in one direction but **in two**.*

From the purely syntactic point of view, it is also possible to allow adjectives and numerals in nominal syntactic positions, and to allow their combination with articles and prepositions. But postulating zero nouns (*the rich people, in two directions*) not only allows us to keep syntax simple, it also makes possible to attach lexical meaning (e. g., 'direction') to a separate syntactic node, making other computations more straightforward.

Example (14) is more difficult for us, because we take prepositions to be children of nouns:

(14) ***Privacy of** and access to information.*

Here again ellipsis greatly simplifies the structure (*[privacy [[of] ~~information~~]]*); but with some ingenuity we could do without it.

2.2. Ellipsis templates

An ellipsis template describes a fragment of syntactic structure. It specifies both linear and hierarchical relations between constituents. It can mention coordination links. It can check grammar value of nodes against grammar expressions. In short, a template can refer to most aspects of the final structure.

One or more nodes in the template bear the label "new". Those are the null elements. If we can insert them in the sentence, so that the resulting structure fits the template, then null elements are created. (Their existence remains hypothetical; they can make their way into the final structure, or they may not.)

We will not describe the syntax of ellipsis templates here, but the basic idea of a linear-and-hierarchical template is certainly familiar to the reader.[4]

A word of caution is in order. Ellipsis is good for people, as it keeps the grammar simple and intuitive; but the real computational gain comes from templates, not from the use of ellipsis per se. While most other grammar rules are binary and local, templates are holistic, and offer a rich view of the context. If the context is wrong, the template can be rejected very fast. An immediate-constituent grammar designed to handle ellipsis would create many false hypotheses which would then die slowly and painfully, littering the syntactic graph for a while.

2.3. Movement rules

Many models of dependency syntax (most notably, the Meaning-Text Model by Mel'chuk) allow non-projective links to capture long-distance dependencies. Consider, for example, the following sentences:

(16) *Куда ты* ***хочешь****, чтобы я пошёл?*

(17) *Одну сумку ему* ***разрешили*** *оставить.*

Our model handles such cases thus:

- The displaced element is attached to the linearly suitable "adoptive parent", which must be an ancestor of its "true" parent,
- It is attached in a special slot that has no semantic interpretation.
- This special slot triggers a rule, which searches the subtree of its "adoptive" parent, looking for a place to put a "proform", or "movement trace".
- A non-tree link is created between the displaced element and its proform.

Walking the syntactic tree from the displaced element to its proform, we go one step up (to the "adoptive parent") and several steps down (to its "true parent", under which the proform is attached). The movement rule specifies the descending part of this path in a **path template**. As in the previous section, we will not discuss the exact syntax of path templates. In a way, they resemble regular expressions.

At one phase of the translation process, a structure is transformed so that all movements are discarded, and all displaced elements return to their "true" parents.

[4] Querying and transforming XML, for instance, requires similar techniques, since the structure of an XML document is both linear and hierarchical, with some non-local references between nodes.

2.4. Syntactic control

A similar mechanism to movement is used to capture control constructions, discussed extensively by the generative grammarians:

(18) *Alice* ***promised*** *Bob to come.*

(19) *Alice* ***persuaded*** *Bob to come.*

(20) *Bob was* ***persuaded*** *to come.*

In this construction, the null subjects of infinitives are controlled either by subject, as in (18) and (20), or by direct object, as in (19). Technically, the infinitive is attached in a slot that triggers the rule of control. This rule has several variants, corresponding to the configurations in (18)–(20). Note that the variants have to check both syntactic grammemes (active vs. passive voice), and classifying grammemes (what type of control is associated with specific verbs).

3. Semantics

3.1. The Semantic Hierarchy

In Compreno project lexical items are organized in the form of a **thesaurus hierarchical tree**. The tree consists of language-independent branches (nodes), called **semantic classes**. These classes are filled with lexical contents in natural human languages (now the descriptions of Russian and English are available, German, French, Chinese — in progress).

Classes of the upper levels are classes denoting general notions — such as entities or actions. Classes of the lower levels represent more particular notions. The resulting taxonomy mostly varies from 3 to 10 levels (unlike "natural" taxonomies that are limited to five levels, according to [1]). The deeper the terminal branch of the taxonomy is the more specific notion it can contain, e. g. *'freely convertable currency'* is located on the 6th level of the hierarchy while 'money' is on the 4th:

FREELY CONVERTABLE CURRENCY < CURRENCY < MONEY < INFORMATION AND SOCIAL OBJECTS < ENTITY < ENTITY-LIKE CLASSES.

The upper-level semantic classes (in this case, INFORMATION AND SOCIAL OBJECTS, ENTITY, ENTITY-LIKE CLASSES) usually contain other semantic classes. The terminal branches (in this case, FREELY CONVERTABLE CURRENCY), by definition, contain no sub-branches, but only "leaves", language-specific **lexical classes** (in this case, English term *'freely convertible currency'*, Russian term *'свободно конвертируемая валюта'* and Russian abbreviation for the term — *'СКВ'*). The intermediate classes (CURRENCY and MONEY) can include both lexical classes and semantic classes: i. e., MONEY includes lexical classes *'деньги — money'* as well

as semantic classes CURRENCY, DEPOSIT, FUND, INTEREST, SAVINGS and so on. Lexical classes adjoining semantic classes (like *'money'* here) are hyperonyms for the classes located below.

Lexical class is often a set of several words with the same root: the lexical class *'деньги'* ('money'), for instance, includes lexemes *'деньги'*, *'денежный'* ('monetary') and *'безденежный'* ('moneyless'). The lexemes *'деньги'* and *'денежный'* differ in syntactic category only, while *'безденежный'* also has evident semantic distinctions. So lexical classes (as well as semantic classes) usually represent not a single meaning, but rather a set of closely related meanings.

All words the hierarchy contains are provided with grammatical and semantical information. We refer to the semantic information units as semantemes by analogy with grammemes (more often the term seme is used for similar purposes).

Semantemes are language-independent meaning elements that perform several functions. **Distributional semantemes**, for instance, are used for grouping classes with similar properties from different branches of the semantic tree which facilitates to describe the semantical compatibility of such classes (i. e., 'plants' and 'metals' both have a <<Substance>> semanteme, 'soup' and 'tears' — semanteme <<Liquid>>). **Differential semantemes** help to differentiate lexical items within one semantic class (*'fat'* has a <<PolarityPlus>> semanteme vs *'thin'* — <<PolarityMinus>>; *'бабки'* (slang word for 'money') differs from neutral *'деньги'* with a <<SocialStatus-Low>> semanteme).

The descendants of one lexical class that differ in semantemes are called **semantic derivates**. The above-mentioned *'безденежный'*, for example, is a semantic derivate of the lexical class *'деньги'* marked with a semanteme <<NotToHave>> and thus is translated as *'moneyless'* with the same semanteme.

There are as well derivates, especially verbal, that are formed by regular morphological models, express the same semantical relations and differ from the 'neutral' derivate in the semantic valencies they can have. For instance, verbs like *'вшить — sew in, вклеить — glue in, ввязать — knit in'* express the semantic relations of contact with another object and localization inside the other object, and are all formed with the same means — 'в-'prefix in Russian and 'in'-particle in English, which influences on the semantic valencies they can have as well.

Such semantic derivates are marked with **derivatemes** — set combinations of corresponding grammemes and semantemes, which helps to describe both the syntactic and the semantic features of these derivates (for more detailed analyses see [9]).

3.2. Semantic slots

The semantic links and relations between words are expressed with the help of **semantic slots**, which, to some extent, correlate with semantic valencies in L. Tesnière's dependency grammar theory [7], deep cases in Ch. Fillmore's case grammar theory [3] or semantic and thematic roles in later linguistic models and conceptions.

An important difference is that the earlier theories as well as later studies generally focus on verbal arguments, underlining the difference between complements and modifiers, while in Compreno project all possible semantic dependencies are taken into account. I.e., there are not only semantic slots corresponding to widely used semantic roles such as [Agent] in *'[the boy] works'* or [Instrument] in *'the letter is written [with a pen]'* but also characteristical slots like [Ch_Evaluation] in *'[beautiful] dress'* or [Ch_Emotion] in *'He looked [surprisingly] exhausted'*, parenthetical slots like [ParentheticalSpecification] in *'you, [for example]'* and plenty of others: [RepresentedFormOfObjectOrCharacteristic]: *'rain [in large drops]'*, [Function]: *'work [as a teacher]'*, [Specifier_Number]: *'gate [1]'* and so on, more than 300 slots in total.

Semantic slots are language-independent objects, like semantic classes, and acquire their surface syntactic realizations in every language. I.e., semantic slots correspond to surface, or syntactic, slots like $Subject, $Object_Direct, $Modifier_Attributive and so on (surface slots are marked with the '$' sign).

The semantic hierarchy is organized according to the **inheritance principle**: many slots are introduced on the upper levels and the child semantic classes inherit them. For instance, locative and temporal adjuncts as well as some characteristical slots like the above-mentioned [Ch_Evaluation] are introduced on the very top of the hierarchy as such constituents can be governed by almost any cores: *'a book [on the table], most important [in the world], working [at school]'*.

Conditional and concessive clauses, in turn, are usually governed by verbal classes only, so the [Condition] and [Concession] slots are introduced on a lower level and cores with entity-like semantics don't inherit them. Constituents with the semantics of motive (such as *'to love smb. [for his talent], to criticize smb. [for wanting to break the rules]'*) or attributes like *'powerful'*, *'twenty-watt'* are attached just to the classes with rather particular semantics, so [Motive] and [Ch_Parameter_Power] slots are introduced even lower.

Another important restriction on the semantical compatibility is a **filling** of the semantic slots: each semantic slot can be filled with a strict set of the semantic classes. I.e., [Agent] is mainly filled with beings, organizations and some territorial units: *'[we/our school/Russia] agrees, that…'*, while [Condition] slot can be filled with any verbal classes.

Slots with similar semantic roles but different fillings are grouped in classes. Thus, [Agent_Class] includes [Agent_Route], [Agent_Device] and [Agent_Metaphoric] besides [Agent] itself. Whereas [Agent] is a slot widely used with different cores, [Agent_Route], for instance, is a slot that mainly verbs of motion have. It is filled with classes like 'ROAD', 'STAIRS', 'RAILWAYS' and other possible 'routes'. Introducing such a slot helps to describe regular metaphors like *'[the stairs] went up, [The railway line] follows the coast'* and to avoid creating corresponding homonyms for the motion verbs in the hierarchy (as it is usually done in most dictionaries).

Another strategy to describe selectional restrictions is using a widely filled slot, which can be introduced on a relatively high level and narrowed lower on some particular classes: i.e., [Object] slot can normally be rather widely filled (*'to see [a boy/a house/somebody's beauty/uncertainty]'*), but some verbs — like *'eat'*, *'drink'*

or *'smoke'* — demand the narrowing of its filling (here is when the above-mentioned distributional semantemes help).

To avoid the 'repairing contexts' problem (compatibility problem occurring in contexts allowing the violations of the selectional restrictions, such as *'I'll eat [my hat] if Kim ate [a motor-bike]'* [5]), we define two sets of fillers for each semantic slot: the allowed one and the preferred one. So when the narrowing is necessary, generally only the preferred fillers are reduced.

4. Disambiguation

In Compreno, we talk of **homonymy** (not distinguished from polysemy) in a situation, when one lexeme belongs to several lexical classes, and of **synonymy** — when one semantic class has several lexical classes with the equivalent set of semantemes.

At the early stages of parsing we build the syntactic tree from lexemes, leaving the semantic ambiguity unresolved as long as possible. By delaying the choice of a specific lexical class, we gain computational efficiency.

The syntactic relations between words can also be rather ambiguous. English nominal premodifiers are a good example: cf. *street fight* and *sword fight* (with "street" and "sword" denoting place and instrument respectively). I.e., **syntactic homonymy**, or polysemy, occurs in a situation when one syntactic relation has several semantic interpretations. **Syntactic synonymy**, in turn, occurs in a situation when one semantic relation has several syntactic realizations (e. g. *John's father = the father of John*).

To deal with this effectively, we distinguish syntactic and semantic relations (through introducing the above-mentioned semantic and syntactic slots), just as we distinguished lexemes and semantic classes. Syntactic relations are language-specific, while semantic relations are language-independent.

At the early stages of parsing the edges of the syntactic graph are labeled with syntactic relations only. The semantic ambiguity is kept unresolved as long as possible. By delaying the choice of a specific semantic relation, we also gain computational efficiency.

References

1. *Cruse D. A.* (1986), Lexical semantics, Cambridge.
2. *Sant-Dizier P., Viegas E.* (1995), An introduction to lexical semantics from a linguistic and a psycholinguistic perspective, in P. Sant-Dizier, E. Viegas (eds.), Computational lexical semantics, Cambridge, pp. 1–29.
3. *Fillmore Ch.* (1968),The case for case, in E. Bach, R. Harms (eds.), Universals in linguistic theory, New York, Holt, Rinehart and Winstonm, pp. 1–90.
4. *Pike K. L.* (1963), A syntactic paradigm, Language, vol. 39, No.2, pp. 216–230.
5. *Soehn J.-Ph.* Selectional Restrictions in HPSG: I'll eat my hat! Proceedings of the HPSG-2005 Conference. University of Lisbon, Portugal. Stanford, CSLI Publications, 2005, pp. 343–353.

6. *Stump G. T.* (2002), Morphological and syntactic paradigms: a theory of paradigm linkage, in G. Booij, J. van Marle (eds.), Yearbook of Morphology. 2001.
7. *Tesnière L.* (1959), Éleménts de syntaxe structurale, Paris, Klincksieck.
8. *Gruntova E. S.* Reguljarnye modeli upravlenija russkih pristavochnyh derivatov [Regular subcategarization frames of Russian prefixal verbs]. Komp'juternaja lingvistika i intellektual'nye tehnologii: Trudy mezhdunarodnoj konferencii «Dialog'2006» [Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2006"]. Moscow, 2006.
9. *Zaliznjak A. A.* (1967), Russkoe imennoe slovoizmenenie [Russian nominal inflexion], Moscow.
10. *Plungjan V. A.* (2011). Vvedenie v grammaticheskuju semantiku: grammaticheskie znachenija i grammaticheskie sistemy jazykov mira [Introduction in grammatical semantics: grammatical values and grammatical systems in languages of the world]. Moscow.

RUSSIAN DEPENDENCY PARSER SYNTAUTOM AT THE DIALOGUE-2012 PARSER EVALUATION TASK

Antonova A. A. (antonova@yandex-team.ru),
Misyurev A. V. (misyurev@yandex-team.ru)

Yandex

In this paper we describe the SyntAutom parser submitted at the Dialogue-2012 parser evaluation task. It is a rule-based system, which performs syntactic and morphological ambiguity resolution in a unified framework. The underlying grammar formalism is Dependency Grammar. The segmentation, shallow parsing and deep parsing are based on a special form of finite-state pushdown automatons, implemented as a sets of recursive functions. The input of the automaton is a lattice of objects — these may be words or shallow trees. Within the functions we have implemented a mechanism of branching and multiple return — a possibility for a function to generate a bunch of states.

We discuss the system architecture, the output parsing trees structure and the common types of incorrect analyses. The distinctive feature of the system is that it tends to directly connect meaningful words, while auxiliary words are demoted to the lower tree levels.

Although the system experiences the common problems of most rule-based systems, it has proven its applicability in a range of real-world applications.

Keywords: natural language parsing, rule-based parser, dependency grammar, Russian syntax

1. Introduction

This paper briefly describes the Russian dependency parser SyntAutom presented at the Dialogue-2011 parser evaluation task. It is a rule-based system, which performs syntactic and morphological ambiguity resolution in a unified framework. The previous versions of the parser are described in [1, 2].

The underlying grammar formalism is Dependency Grammar[3, 5, 6, 8]. The search algorithm is bottom-up and depth-first. The parsing strategy is based on a special form of finite-state pushdown automaton, implemented as a set of recursive functions. The input of the automaton is a lattice of objects — these may be words or shallow trees. An automaton state has access to the current position in the sentence, the global parameters(e. g. the closeness of the sentence boundary, the syntactic class of the previous independent tree), the parameters of the last read object and the parameters of all objects in the stack. Possible transitions to other states are specified by parsing rules, based on whether the current state parameters satisfy the certain conditions.

We have developed a native formalism for writing parsing rules in form of functions. Each function has a predefined scope — objects and parameters that the function can manipulate. All functions also have access to global parameters. Within functions we have implemented a mechanism of branching and multiple return — a possibility for a function to generate a bunch of states. All newly created states are then processed independently.

An automaton path is a sequence of states, which corresponds to a contiguous fragment of the input sentence. Each path reveals dependencies between words. Due to the morphological and syntactic ambiguity many different paths can be associated with the input sentence or its fragment. When all paths are explored, the parsing tree is reconstructed according to the dependencies found along the best path.

The automaton does not allow to skip any words. Obviously one cannot expect to find a full parse tree for all the unrestricted variety of natural sentences. So the common convention is to allow the sentence to break on several parts and look for the best partial parses. The pushdown automaton can split the sentence every time its stack is empty. Then it begins parsing from the next word as if there was a sentence boundary before it. Different paths can put partial sentence boundaries in different positions.

We describe the overall structure of the system in Section 2. In Section 3 we specify the parser output format and explain representation of some specific syntactic structures. We discuss the limitations of our parsing approach in Section 4. Section 5 describes applications of the system. We conclude in Section 6.

2. System architecture

As a rule-based system, SyntAutom relies on hand-built parsing rules and morphological dictionary, as well as some additional data sources. The parsing process is represented in Figure 1.

2.1. Segmentation and morphological analysis

SyntAutom has a sophisticated segmentation procedure, which is responsible for the detection of:

- sentence boundaries (if the input is not sentence-splitted);
- phrases that function as single words;
- complex tokens (e. g. e-mail, url);
- proper names;
- compound cardinal numbers;
- hyphenation.

The segmentation procedure also represents a finite-state automaton, implemented as a set of functions. The automaton receives a string of low-level textual tokens as input. The output of this stage is a lattice, representing different segmentation

options, morphological and structural ambiguities, including possible multiword expressions. Due to morphological ambiguity a surface word can be associated with different lattice nodes, having different lemmas or different sets of morphological features. We call such nodes morphological interpretations. Each morphological interpretation is assigned a set of lexical features (e. g. from morphological dictionary or valence list). Unknown words are analyzed heuristically, by finding similar words in morphological dictionary.

The texts provided by the Dialogue-2011 parser evaluation task had been sentence-splitted and tokenized by the organizers. That probably reduced the number of certain segmentation mistakes.

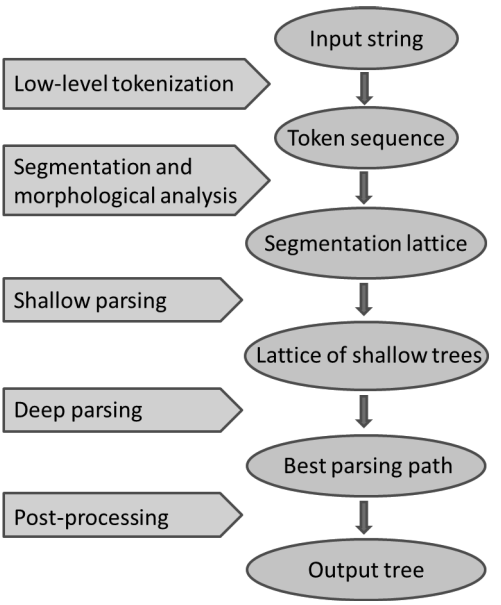


Figure 1. The system architecture

2.2. Verb valences

Typically, morphological dictionaries do not contain information about verb valences. At the same time good-quality valence information is crucial for the performance of a rule-based system. We manually created lists of verb valences for more than 12'000 Russian verbs. The following valences for a verb are indicated:

1) a subject in nominative case;
2) an object in accusative case;
3) an object in dative case;
4) an object in genitive case;
5) an object in instrumental case (optional);

6) a subordinate clause, beginning with “что”(“that”);
7) a subordinate clause, beginning with question words (“где”(*“where”),* “когда”(“*when*”), “как”(“*how*”), “почему”(“*why*”), “какой”(“*which*”), “чей”(“*whose*”));
8) possibility to be auxiliary for another verb (see section 3.4)

Some valence combinations are mutually exclusive. We provide each verb with possible combinations of the 8 valence types, instead of indicating each valence independently. For example, possible combinations for the verb “жалеть” (“be sorry”) are [1,2], [1,4], [1,6]; “угрожать” (threaten) — [1,3,5], [1,3,6]. The same logic can be applied to adjectives and nouns, some of which can also have special valences.

2.3. Shallow parsing

At the shallow parsing stage the system tries to detect simple phrasal groups that cannot have recursive structure. We do not use pushdown operations on this stage. The shallow parsing strategy can be described as a simple finite-state automaton with right-to-left expansion[1].

Examples of shallow trees that can be constructed on this stage:

1. Noun phrase with left modifiers (except for left participle when it has its own complements).
   “большой дом” (“*big house*”);
   “большой красивый дом” (“*big beautiful house*”);
   “очень большой дом” (“*very big house*”);
   “двадцать два больших дома” (“*twenty two big houses*”);
   “с большим домом” (“*with a big house*”);
2. Adjectival or verb phrase with left modifiers.
   “очень быстрый” (“*very fast*”);
   “быстро бежал” (“*ran fast*”, literally “*fast ran*”);
   “очень быстро бежал” (“*ran very fast*”, literally “*very fast ran*”);

The output of the shallow parsing is a lattice of shallow trees, and it is the input to the deep parsing automaton. The shallow parsing finds rather simple phrases, but it helps to save time for the computationally heavy task of deep parsing.

2.4. Deep Parsing

Deep parsing deals with long-distance dependencies and embedded structures - that is why pushdown operations(stack) are necessary here. The deep parser strategy

[1] Since we mostly explore left modifiers, at this stage the automaton reads words from right to left.

is bottom-up, depth-first, with left-to-right expansion. The parser performs non-deterministic, exhaustive search. It does not attempt to resolve each decision as reached, but rather pursues all alternatives. The parser cannot skip words. Even punctuation and unknown words must have their place in the tree.

An important optimization, that allows to avoid repeated analyses and reduce combinatorics, is caching the function calls together with the parameters that belong to the scope of this function. If another function call happens to have exactly the same parameters, the parser simply retrieves the resulting bunch of states from the cache, without actually running the function. A similar approach, called «chart parsing» is described in [7].

At the end of the deep parsing stage, the path with the best weight is chosen. The evaluation of an automaton path weight is performed based on the following factors:

1) Frequencies of morphological interpretations of words.
2) Frequencies of binary lexicalized dependency relations.
3) Empirical weights that are added when the path crosses some state in the automaton.

As in most rule-based systems, empirical weights are important for the system performance. They allow to penalize or promote some automaton states according to the linguistic intuition. But the roughness and inaccurateness of the empirical weights limits the usefulness of any statistical factors. Thus, we used the simple add-one smoothing when evaluating the frequencies of different morphological interpretations of words and dependency relations.

The frequencies of binary lexicalized dependencies were extracted from large automatically parsed corpus. Though there definitely exist many parsing errors in the corpus, the overall performance slightly improved. For example, in some cases such statistics helps to resolve the problem of prepositional phrase attachment.

Мне нравилось смотреть на улицу через стекло

(I liked to look at the street through the glass)

0	*Top*	*Top*	0	_	/
1	мне	я	3	subj	/prn/sg/fem/msc/neu/dat/fst/
2	нравилось	нравиться	3	auxd	/vrb/sg/neu/fin/fst/sec/trd/pst/ind/act/
3	смотреть	смотреть	0	fin	/vrb/sg/neu/inf/fst/sec/trd/pst/act/
4	на	на	5	prep	/prp/acc/
5	улицу	улица	3	prepnp	/nn/sg/fem/acc/trd/
6	через	через	7	prep	/prp/acc/
7	стекло	стекло	3	prepnp	/nn/sg/neu/acc/trd/

2.5. Post-Processing

A post-processing stage was added to improve parser performance for the evaluation task.

Adverbial participles were made dependent on the head of neighboring clause. Clauses beginning with a conjunction were made dependent on the head of neighboring clause.

We have also tried to redirect certain types of dependency relations, to avoid disagreement with the gold standard parses. That included prepositions, cardinal numbers and modal verbs. We did not redirect dependency relations in cases when the dependent word had its own dependents.

3. Output format and tree structure

In the rest of the paper the parser results are represented as a table. Each line corresponds to one word of the input sentence and consists of:

- word id;
- word form;
- lemma;
- id of the head word;
- tag of the dependency relation;
- morphological attributes.

The words are enumerated according to the order in the input sentence.

An artificial word "*Top*" precedes each sentence. It acts as a head word for words which have no other parent, enforcing a single-tree structure even for partial parses. In some cases the segmentation module inserts additional "*Top*" in the middle of the sentence.

в свои родные края, к своей работе, завести детей,
прожить долгую полноценную жизнь.
(To his homeland, to his work, to have children,
to live a long full life)

0	*Top*	*Top*	0	_	/
1	в	в	4	prep	/prp/acc/
2	свои	свой	4	adj	/prn/pl/msc/nom/trd/
3	родные	родной	4	adj	/adj/pl/msc/nom/trd/
4	края	край	0	prepnp	/nn/pl/msc/acc/trd/
5	,	,	0	misc	/pnt/
6	к	к	8	prep	/prp/dat/
7	своей	свой	8	adj	/prn/sg/fem/loc/trd/
8	работе	работа	0	prepnp	/nn/sg/fem/dat/trd/
9	,	,	0	misc	/pnt/
10	завести	завести	0	inf	/vrb/inf/act/
11	детей	ребенок	10	acc	/nn/pl/msc/anm/acc/trd/
12	,	,	10	conj	/pnt/
13	прожить	прожить	10	homo	/vrb/inf/act/
14	долгую	долгий	16	adj	/adj/sg/fem/acc/trd/

15	полноценную	полноценный	16	adj	/adj/sg/fem/acc/trd/
16	жизнь	жизнь	13	acc	/nn/sg/fem/acc/trd/
17	.	.	0	misc	/pnt/

There are several constraints on the general tree structure. Every word, even punctuation and unknown words, must find their place in the tree. In case of a partial parse tree, no dependencies are allowed across an independent part of the sentence. Punctuation at the end of the sentence is usually dependent on "*Top*", except when it is recognized as a part of an abbreviation.

A dependency relation tag can be considered to be a syntactic role of the word with respect to its head word. If a word is the head of an independent tree, then its tag corresponds to the syntactic class of the tree. The set of syntactic roles and classes is described in Appendix.

Our system follows the common practical convention that subject, object and other complements are dependent on the verb.

моя сестра подарила мне эти жемчужины

(My sister gave me these pearls)

0	*Top*	*Top*	0	_	/
1	моя	мой	2	adj	/prn/sg/fem/nom/trd/
2	сестра	сестра	3	subj	/nn/sg/fem/anm/nom/trd/
3	подарила	подарить	0	fin	/vrb/sg/fem/fin/trd/pst/ind/act/
4	мне	я	3	dat	/prn/sg/fem/msc/neu/dat/fst/
5	эти	этот	6	adj	/prn/pl/fem/nom/trd/
6	жемчужины	жемчужина	3	acc	/nn/pl/fem/acc/trd/

Still there are some constructions for which our analysis may differ from analyses of other systems. Although we do not pretend to explore semantic relations in the sentence, there is a number of situations when we prefer to mark a semantic dependency instead of a syntactic one.

3.1. Prepositions

We consider preposition to be a dependent on the noun phrase rather than its head. Prepositions often convey little or no meaning, whereas in many tasks it is helpful to have a direct dependency between meaningful words.

он был женат на креолке из евангелического прихода

(he was married to a creole from the evangelical parish)

0	*Top*	*Top*	0	_	
1	он	он	3	subj	prn/sg/msc/nom/trd
2	был	быть	3	auxs	vrb/sg/msc/fin/fst/sec/ trd/pst/ind/act
3	женат	женатый	0	fin	adj/sg/msc/trd/pst/sht

4	на	на	5	prep	prp/loc
5	креолке	креолка	3	prepnp	nn/sg/fem/anm/loc/trd
6	из	из	8	prep	prp/gen
7	евангелического	евангелический	8	adj	adj/sg/msc/gen/trd
8	прихода	приход	5	prepnp	nn/sg/msc/gen/trd

3.2. Coordination

First coordination member is always the representative of the coordination group. Each successive coordination member is attached to the previous member with the dependency relation “homo”. The coordinating conjunction or comma is also attached to the previous member with the dependency relation “conj”.

чтобы чинить, оказывать помощь и спасать

(to repair, to render assistance and rescue)

0	*Top*	*Top*	0	_	/
1	чтобы	чтобы	0	conj	/cnj/
2	чинить	чинить	0	inf	/vrb/inf/act/
3	,	,	2	conj	/pnt/
4	оказывать	оказывать	2	homo	/vrb/inf/act/
5	помощь	помощь	4	acc	/nn/sg/fem/acc/trd/
6	и	и	4	conj	/cnj/
7	спасать	спасать	4	homo	/vrb/inf/act/

3.3. Auxiliary and modal verbs

We consider as auxiliary almost all instances of verb usage when two verbs are syntactically related and subject of both verbs is identifiable. The parser does not distinguish between auxiliary and modal verbs, but rather makes a distinction based on whether the semantic subject of the verbs is the same or different.

я хотела бы оказаться там

(I would like to be there)

0	*Top*	*Top*	0	_	/
1	я	я	4	subj	/prn/sg/fem/nom/fst/
2	хотела	хотеть	4	auxs	/vrb/sg/fem/fin/fst/sec/trd/pst/ind/act/
3	бы	бы	2	by	/pt/
4	оказаться	оказаться	0	fin	/vrb/sg/fem/inf/fst/pst/act/
5	там	там	4	adv	/adv/

Here the semantic subject “я”(“*I*”) is the same for both the main and the modal verb, and the modal verb “хотеть”(“*want*”) has the syntactic role “auxs”. In this case the subject is made dependent on the main verb, rather than on the modal verb.

она научила меня играть на пианино
(She taught me to play the piano)

0	*Top*	*Top*	0	_	/
1	она	она	2	subj	/prn/sg/fem/nom/trd/
2	научила	научить	4	auxd	/vrb/sg/fem/fin/trd/pst/ind/act/
3	меня	я	4	subj	/prn/sg/fem/msc/neu/acc/fst/
4	играть	играть	0	fin	/vrb/sg/fem/inf/fst/sec/trd/pst/act/
5	на	на	6	prep	/prp/acc/loc/
6	пианино	пианино	4	prepnp	/nn/sg/neu/acc/trd/

Here the semantic subject of the main verb “играть”(“*to play*”) is different from the subject of the modal verb “научить”(“*teach*”), and the modal verb has the syntactic role “*auxd*”. In this case both semantic subjects are dependent on the corresponding verbs with the role “*subj*”. It is worth to note that other arguments are usually dependent on the main verb, rather than on the modal verb.

3.4. Subordinate clauses

The head predicate of a subordinate clause is made dependent of the head predicate of the main clause with the dependency relation “sent”.

он рассказывал что там играл духовой оркестр
(He told that there were a brass band)

0	*Top*	*Top*	0	_	/
1	он	он	2	subj	/prn/sg/msc/nom/trd/
2	рассказывал	рассказывать	0	fin	/vrb/sg/msc/fin/trd/pst/ind/act/
3	что	что	5	conj	/cnj/
4	там	там	5	adv	/adv/
5	играл	играть	2	sent	/vrb/sg/msc/fin/trd/pst/ind/act/
6	духовой	духовой	7	adj	/adj/sg/msc/nom/trd/
7	оркестр	оркестр	5	subj	/nn/sg/msc/nom/trd/

3.5. Cardinal numbers

We consider a cardinal number to be a dependent on the noun phrase.

У нее было двадцать две кошки.
(She had twenty two cats)

0	*Top*	*Top*	0	_	/
1	у	у	2	prep	/prp/gen/
2	нее	она	3	prepnp	/prn/sg/fem/gen/trd/
3	было	быть	0	fin	/vrb/sg/neu/fin/fst/sec/trd/pst/ind/act/
4	двадцать	двадцать	5	card	/num/pl/fem/msc/neu/nom/trd/

5	две	два	6	card	/num/pl/fem/nom/trd/
6	кошки	кошка	3	subj	/nn/pl/fem/anm/nom/trd/

4. Robustness vs. coverage

In the real-world applications the parser robustness is more important than its "grammatical coverage" – the system's theoretical ability to build certain types of parse trees. The more permissive is the grammar, the bigger is the chance that the system gets confused with all the possible parses. For example, in a machine-translation application it is often better to translate a fragment word-by-word than to reorder it based on a wrong parse.

We intentionally do not attempt to interpret sentences without predicates: "у меня температура" ("*I <have> a fever*"), "это собака" ("*this <is> a dog*"), "она адвокат" ("*she <is> a lawyer*") "сегодня ты один" ("*today you <are> alone*"), "как ее дыхание?" ("*How <is> her breath?*") "мы больше не друзья" ("*we <are> not friends anymore*"). Sentences without predicates are parsed disconnectedly.

У меня температура
(I have a fever)

0	*Top*	*Top*	0	_	/
1	у	у	2	prep	/prp/gen/
2	меня	я	0	prepnp	/prn/sg/fem/msc/neu/gen/fst/
3	температура	температура	0	np	/nn/sg/fem/nom/trd/

Here we obviously sacrifice some of the potential recall, but eliminate spurious analyses and prevent increase in combinatorics: e.g. the sentence "У меня температура зашкаливает" ("*My temperature is very high*").

The parser does not find connections for some phrases that are not structural parts of the predications, like interjections or direct address construction.

ага, я так и знал
(yeah, I knew it)

0	*Top*	*Top*	0	_	/
1	ага	ага	0	np	/nn/sg/msc/nom/trd/
2	,	,	0	misc	/pnt/
3	я	я	6	subj	/prn/sg/msc/nom/fst/
4	так	так	6	adv	/adv/
5	и	и	4	misc	/cnj/
6	знал	знать	0	fin	/vrb/sg/msc/fin/fst/pst/ind/act/

ты немного староват для войны, бенджамин
(You are a bit old for the war, benjamin)

0	*Top*	*Top*	0	_	/
1	ты	ты	3	subj	/prn/sg/msc/nom/sec/

2	немного	немного	3	adv	/adv/
3	староват	староватый	0	krat	/adj/sg/msc/sec/prs/sht/
4	для	для	5	prep	/prp/gen/
5	войны	война	3	prepnp	/nn/sg/fem/gen/trd/
6	,	,	0	misc	/pnt/
7	бенджамин	бенджамин	0	np	/nn/sg/msc/anm/nom/trd/cap/

Nevertheless, the big number of possible analyses is still an important problem. There exist constructions the addition of which can increase combinatorics dramatically. For example, constructions with emphatic “и”(*and*), the addition of which can complicate the recognition of coordination constructions. The constructions with emphatic “и”(*and*) are parsed disconnectedly in the current version of the parser.

Пройдет и это

(This will pass too)

0	*Top*	*Top*	0	_	/
1	пройдет	пройти	0	fin	/vrb/sg/fem/msc/neu/fin/trd/prs/ind/act/
2	и	и	0	conj	/cnj/
3	это	этот	0	np	/prn/sg/neu/acc/trd/

As a rule-based system our parser relies heavily on the word valence information. We do not use prepositional valencies - any prepositional group can depend on any previous noun phrase or verb phrase. The system allows any verb to have a complement in instrumental case, though verbs that have a predefined instrumental valence are encouraged. The ability to govern other non-prepositional complements(genitive, dative, accusative), subject and subordinate clauses is strictly controlled by the predefined valence information. For that reason the parser is unable to recognize a dependency if it is not permitted in its valence lists. This kind of mistakes can be divided into two subtypes:

1. Mistakes that can be corrected easily if we add a missing valence to the valence lists.

вещают мортимеру про конец

(They prophecy to Mortimer about the end)

0	*Top*	*Top*	0	_	/
1	вещают	вещать	0	fin	/vrb/pl/fem/msc/neu/fin/trd/prs/ind/act/
2	мортимеру	мортимер	0	np	/nn/sg/msc/anm/dat/trd/
3	про	про	4	prep	/prp/acc/
4	конец	конец	2	prepnp	/nn/sg/msc/acc/trd/

Here we can add a missing dative valence to the verb “вещать” (“*to prophecy*”) and improve the parser performance.

2. Mistakes in which a missing valence is lexically-dependent, for example, Russian construction with dative possessor.

он поцеловал невесте руку.
(He kissed the bride's hand)

0	*Top*	*Top*	0	_	/
1	он	он	2	subj	/prn/sg/msc/nom/trd/
2	поцеловал	поцеловать	0	fin	/vrb/sg/msc/fin/trd/pst/ind/act/
3	невесте	невеста	0	np	/nn/sg/fem/anm/loc/trd/
4	руку	рука	0	np	/nn/sg/fem/acc/trd/

Here we cannot add a dative valence to the verb «поцеловать», because it depends on the semantics of the other complement (part of body). The parser usually cannot parse this type of constructions correctly.

We allow contextual substantivation of adjectives - i.e. each adjective can act as noun phrase in many contexts.

коричневый идет вашим глазам
(The brown suits your eyes)

0	*Top*	*Top*	0	_	/
1	коричневый	коричневый	2	subj	/adj/sg/msc/nom/trd/
2	идет	идти	0	fin	/vrb/sg/msc/fin/trd/prs/ind/act/
3	вашим	ваш	4	adj	/prn/pl/msc/dat/trd/
4	глазам	глаз	2	dat	/nn/pl/msc/dat/trd/

Accusative-genitive case transformation in negative sentences. "Я вижу собаку" *(I see the dog)*, "Я не вижу собаки" *(I don't see the dog*. We assign grammatical role "acc" in both cases.

Я не вижу собаки
(I don't see the dog)

0	*Top*	*Top*	0	_	/
1	я	я	3	subj	/prn/sg/fem/msc/neu/nom/fst/
2	не	не	3	pt	/pt/
3	вижу	видеть	0	fin	/vrb/sg/fem/msc/neu/fin/fst/prs/ind/act/
4	собаки	собака	3	acc	/nn/sg/fem/anm/gen/trd/

5. Practical applications

Initial application of our parser was within a rule-based machine translation system. The parser has already been used as a tool for preparation and analysis of linguistic corpora. For example, the automatically created treebank of 70 million sentences had been used for the lexicographic purposes.

1) Automatic creation of bilingual dictionary. The parsing information was used to filter out ungrammatical phrases and find words and phrases in canonical form.

2) Automatic extraction of synonyms. The parsing information was used to compute distributional similarity between words.

Other applications include distributional semantic clustering, context classification, text comparison (in particular [9, 10]). The parser was also used in student works of Moscow State University (in particular [11]).

Conclusion

We have described the dependency parser submitted at the Dialogue-2011 parser evaluation task. The details of the output tree structures and the common types of incorrect analyses were discussed. The distinctive feature of the system is that it tends to directly connect meaningful words, while auxiliary words are demoted to the lower tree levels.

The advantages of our formalism are the following:
1) The syntactic and morphological ambiguity is resolved simultaneously within a unified framework.
2) The explicit description of automaton transitions provides a flexible way to control parsing process.
3) The automaton state usually provides more information than a context-free rule can provide.
4) It is easy to add "local" functions that are called only in specific conditions.

The system experiences the common problems of most rule-based systems. First, it is difficult to reconcile empirical weights with the weights provided by statistical models. Second, it is hard to increase grammatical coverage beyond certain extent, because of the increase in combinatorics and the drop in precision.

In spite of existing limitations, the system has proven its applicability in a range of real-world applications.

References

1. *Antonova A. A., Misyurev A. V.* The development of a syntactic parser for Russian and English [Realizatsija sintaksicheskogo razbora dlja russkogo i anglijskogo jazykov], Pervaja Mezhdunarodnaja konferentsija "Sistemnyj analiz i informatsionnye tehnologii" [The first international conference "System analysis and information technologies"], Pereslavl'-Zalesskij, 2005, pp. 245–249.
2. *Antonova A. A., Misyurev A. V.* (2008), About applications of the syntactic parser Cognitive Dwarf 2.0. [Ob ispol'zovanii sintaksicheskogo analizatora Cognitive Dwarf 2.0.], Sbornik trudov ISA RAN [Proceedings of ISA RAS], no. 38, pp. 91–109.

3. *Jurij Apresian, Igor Boguslavsky, Leonid Iomdin, Alexander Lazursky, Vladimir Sannikov, Victor Sizov, Leonid Tsinman* (2003) ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT, MTT 2003. First International Conference on Meaning-Text Theory Paris, Ecole Normale Superieure, pp. 279–288
4. *Gershenzon L. M., Nozhov I. M., Pankratov D. V., Sokirko A. V.* Syntactic analysis in RML system. [Sintaksicheskij analiz v sisteme RML], available at http://www.aot.ru/docs/synan.html
5. *David Hays G.* (1964). Dependency theory: A formalism and some observations. Language, 40: P. 511–525.
6. *Hudson Richard* (1991) English Word Grammar. Basil Blackwell, Cambridge, MA.
7. *Kay, Martin.* (1986). Algorithm schemata and data structures in syntactic processing. Readings in Natural Language Processing, pp. 35–70. Los Altos, CA: Morgan Kaufmann.
8. *Mel'ˇcuk Igor A.* (1987) Dependency Syntax: Theory and Practice. State University of New York Press, Albany.
9. *Mihajlov D. V., Emeljanov G. M.* (2010). Theoretical basis of the development of open-source question-answering systems. Semantic equivalence of texts and recognition models [Teoreticheskie osnovy postroenija otkrytyh voprosno-otvetnyh sistem. Semanticheskaja èkvivalentnost' tekstov i modeli ih raspoznovanija]. NovSU, Velikij Novgorod.
10. *Mihajlov D. V., Emeljanov G. M.* (2011). The analysis of formal concepts and the compression of textual information in the task of automated knowledge control [Analiz formal'nyh ponjatij i szhatie tekstovoj informatsii v zadache avtomatizirovannogo kontrolja znanij]. Matematicheskie metody raspoznavanija obrazov "MMPR-15" [Mathematical methods of pattern recognition], available at http://www.machinelearning.ru/wiki/images/4/4c/Mmpr15_mdv_report.pdf
11. *Tretjakov D., Il'jushina E. A., Puchkova E. M.* (2009) Automated analysis of syntactic relations in Russian texts using Markov Models [Avtomatizirovannyj analiz sintaksicheskih otnoshenij v russkom tekste s ispol'zovaniem tsepej Markova]. Sovremennye problem gazovoj i volnovoj dinamiki [Modern problems of Gas and Wave Dynamics], MSU, Moscow, p.107.

Appendix

Russian syntactic roles

subj	subject
acc	direct object
dat	dative-case object
ins	instrumental-case object
gen	genitive-case object
prepnp	prepositional phrase

adj	adjectival modifier with no heavy dependent nodes
ptp	adjectival modifier having its own dependent nodes or located after noun
adv	adverb
prep	preposition (depends on a noun)
conj	conjunction
digit	number
card	cardinal numeral
auxs	auxiliary with the same subject
auxd	auxiliary with different subject
inf	infinitive (depends on a verb)
sent	subordinate clause
by	particle “бы”
li	particle “ли”
pt	particle
emph	emphatic conjunction “и”
homo	conjunct
sharp	part of a compound word
hyph	part of a hyphenated word
quoml	left quotation mark
quomr	right quotation mark
misc	other

Top-level syntactic classes

sent	sentence
np	noun phrase
prepnp	prepositional phrase
prep	preposition
adj	adjective/participle phrase
adv	adverbial phrase
fin	finite verb phrase
krat	short-form adjective/participle phrase
inf	infinitive phrase
dee	adverbial participle phrase
imper	imperative verb phrase
misc	other

ETAP PARSER: STATE OF THE ART[1]

Iomdin L. (iomdin@iitp.ru),
Petrochenkov V. (petrochenkvov@iitp.ru),
Sizov V. (sizov@iitp.ru),
Tsinman L. (cinman@iitp.ru)

Institute of Information Transmission Problems
(Kharkevich Institute), Russian Academy of Sciences

The state of the art of the ETAP-3 syntactic parser, which took part in a recent competition of Russian parsers, is presented. The paper gives an outline of the main linguistic resources involved in the parser's operation, describes the main features and steps of the algorithm, and briefly discusses the applications in which the parser is used, including a machine translation system, a software environment for the creation of a syntactically tagged corpus of Russian, and a hybrid system of Russian speech synthesis. Special attention is given to concrete scientific approaches and solutions that determine the functioning of the parser, including methods of lexical and syntactic disambiguation.[2]

Key words: parser, combinatorial dictionary, syntagm, tagged text corpora

1. General information

The syntactic parser presented here is the central component of the ETAP-3 multipurpose linguistic processor, designed and developed at the Laboratory of computational linguistics of the Institute for Information Transmission Problems in Moscow.[3] It has two major options operating on very similar (although not identical) principles: the parser of Russian and the parser of English. In what follows, only the parser of Russian will be described.

The parser (to be henceforth called ETAP, for short) is rule-based, with some statistical components incorporated recently.

ETAP processes the text sentence by sentence and has several modes of operation:

---

[1] The author is grateful to the Russian Foundation of Basic Research, who supported the research upon which this paper is based with grants No.10-06-00478-a and 11-06-00405-a, and to the Presidium of the Russian Academy of Sciences, who supported this study with a Basic Studies Programme on Corpus Linguistics.

[2] Доклад публикуется в сокращенном варианте. Полная версия доступна на сайте конференции «Диалог»

[3] Earlier versions of the parser, as well as individual aspects of its performance and maintenance. were described in detail in Apresjan et al 1989,1992, 2003, Boguslavsky et al 2008, 2011.

- fully automatic mode, applied by default: in this case, only one syntactic structure is built for any sentence processed;
- multiple parsing mode, in which the user may instruct the system to build, for an ambiguous sentence, several syntactic structures or even all possible structures;
- interactive mode, in which ETAP stops at certain points of the algorithm if it encounters an ambiguous lexical unit or syntactic construction. In this case, the user is asked to prompt the system for a morphological, lexical, or syntactic interpretation of the ambiguous element of the sentence and in this way direct the algorithm to take some concrete path.

The ETAP parser is primarily aimed at processing texts of neutral genres (journalism, popular science texts, news messages and the like). It cannot be used to adequately handle colloquial speech, fiction, or poetry, as well as "dirty" texts full of tables, lists, or indexes, as well as texts that are essentially deviant from the Russian literally norm.

1.1. Major linguistic conventions

The linguistic formalism used in ETAP is dependency grammar, and the structures produced are dependency tree structures. To a large degree, it is based on the Meaning ⇔ Text linguistic theory by Igor Mel'čuk, particularly on its surface syntactic component (see e. g. Mel'čuk 1974/1999). ETAP operates with written text, constructing a dependency tree for each sentence of it in turn. As a rule, every node corresponds to one word of the sentence. Punctuation marks do not constitute any nodes and are generally attached to the words preceding them). In certain cases a node can correspond to a string of words, which is treated as an indivisible word for linguistic and/or algorithm optimization reasons.

The arcs of the tree are labeled with names of surface syntactic relations (SyntR). These names indicate the different types of syntactic links between the words. In the current version of the parser, about 70 SyntRs are used. To give a few basic examples,

- the link between a predicate, expressed by a finite verb, which is the head, and its subject, which is the dependent, as in *отец* ← *получил* 'father received', is represented with **predicative SyntR**;
- the link going from a predicate word (verb, noun, adjective, or adverb) to the word instantiating its first complement, as in *получил* → *письмо* 'received a letter', *получение* → *письма* 'reception of a letter', *эквивалентный* → *отказу* 'equivalent to a refusal', *вглубь* → *леса* 'deep into the wood' etc. is represented by the **1st completive SyntR**;
- the link attaching the nominal part of the predicate to the copula verb, as in *был* → *зол* 'was angry' or *будучи* → *учителем* 'being a teacher' is represented by the **copulative SyntR**;
- the link connecting a noun and its adjectival modifier, as in *заказное* ← *письмо* 'registered letter', is represented by the **modificative** SyntR;

- the **adverbial SyntR** is used to represent modifiers of verbs expressed by adverbs or prepositional phrases, as in *неожиданно* ← *получил* 'unexpectedly received' or *получил* → *в понедельник* 'received on Monday';
- analytic forms of words (future tense or subjunctive mood of verbs and comparative degrees of adjectives and adverbs) are considered as syntactic constructions and represented with the help of the **analytic SyntR** (*будет* → *читать* 'will read', *читал* → *бы* 'would read', *более* ← *интересный* 'more interesting');
- the link between a noun and a numeral that refers to it is represented with **quantitative SyntR**, as in *два* ← *стола* 'two tables', *пятерыми* ← *лингвистами* 'by five linguists'. Importantly, the link always points to the numeral[4];
- coordination is rendered on a par with subordination; coordination strings are presented in such a way that the first conjunct is the head on which the second conjunct depends and so on; a coordinating conjunction is subordinated by the conjunct preceding it. In most cases, two syntactic relations are used: the **coordinative SyntR** that links the neighboring conjuncts from left to right, and the **coordinative-conjunctive SyntR** that appends a conjunct to the left-adjacent conjunction.

It should be emphasized that in the course of structure generation ETAP does not produce any additional nodes for words physically absent from the sentence. In particular, no anaphoric pronouns are introduced in sentences like (1) *Иван сказал, что устал* (lit. *Ivan said that was tired,* in which some parsers may add the pronoun *он* 'he'), no elliptic omissions, as in (2) *Я заказал сок, а он пиво* 'I ordered a juice and he a beer' are restored[5]. Moreover, the parser does not even generate special nodes for zero forms of the present tense of the verb *быть* 'to be' (irrespective of its particular lexical meaning), which are so common in Russian. Accordingly, the constructions like (3) *Он был счастлив* 'he was happy' or (4) *Я буду в отпуске* 'I will be on leave' receive parses noticeably different from those generated for sentences like (3a) *Он счастлив* 'he is happy' or (4a) *Я в отпуске* 'I am on leave': compare e. g. parses for (3) and (3a) below:

(3) *он* ◄—predicative—*был*—copulative—► *счастлив;*

(3a) *он* ◄—predicative—*счастлив.*

In some of the applications in which the ETAP parser is used, zero copulas are generated at a later stage of sentence processing.

---

[4] Notwithstanding a widely accepted viewpoint that in the nominative/accusative case of the quantitative NP it is the numeral that controls the noun requiring that it should appear in the genitive.

[5] In the SynTagRus treebank (see below) created with the help of the ETAP parser, elliptic omissions are restored manually. E.g. the parse for (2) receives another node for *заказал*, which is assigned a special feature PHANTOM.

The syntactic tree of the sentence as generated by the ETAP parser is **ordered**: it retains the information on word ordering of the source sentence.

1.2. Major linguistic resources used

ETAP parser makes use of the following two major types of linguistic resources:

- **the grammar**, which consists of several hundreds of binary syntactic rules, or syntagms, and
- **the dictionary.** ETAP resorts to the so-called combinatorial dictionary that contains rich and diverse information on every lexical entry. Conceptually, the combinatorial dictionary can be considered as a simplified version of the explanatory combinatorial dictionary of the Meaning ⇔ Text theory, the main difference being that the ETAP dictionary has no explicit lexicographic definitions. At the moment, the combinatorial dictionary has 100,000 entries.

To illustrate both types of resources, we will briefly describe a syntagm and a dictionary entry.

1.2.1. An example of an ETAP syntagm

The following syntagm, reproduced in Fig.1, is used to generate the predicative link between the verb in the imperative [X] and its subject in the nominative [Y]: this is a construction peculiar for a specific type of Russian conditional sentences like

(5) *Приди* [X] *он* [Y] *раньше, мы бы успели все обсудить* 'If he came earlier (lit. $\text{Come}_{\text{imper}}$.he earlier…) we would have time to discuss all'.

```
REG:ПРЕДИК.05
N:01
CHECK
1.1 =(X,ПОВ,ЕД)
1.2 R-EQU(X,Y,4,ИМ)/LEXR(X,БЫТЬ)& R-EQU(X,Y,4,РОД)&L-EQU(X,*,0,НЕ1)
2.1 PININT(X,Y,ЗПТ,1)
3.1 DEP-EQUN(X,Z,ОБСТ,ЛИЧ,ИНФ)
3.2 DOM-LEXR(Z,*,АНАЛИТ,БЫ)
3.3 PININT(X,Z,ЗПТ,1)/PININT(Z,X,ЗПТ,1)
DO
1 SVUZOT:(X,Y,ПРЕДИК)
```

Fig. 1. A predicative syntagm of ETAP

Syntagms, as all other rules of ETAP, are written in a special formal language for linguistic descriptions, called FORET, based on three-valued first order predicate logic. Somewhat simplifying the picture, we may say that any syntagm consists of two zones: (i) the CHECK zone, which lists the conditions to be verified written with the help of **predicates**, and (ii) the DO zone, which contains an **instruction** to the

algorithm to create a hypothetical syntactic link; this instruction is to be performed if all the conditions of the CHECK zone are satisfied.

All conditions are written in the disjunctive normal form and arranged into several groups, identified by the first figure in the two-position number of the condition. Items belonging to the groups where this figure is odd describe **necessary** conditions that have to be satisfied in order for the syntagm to be applied, and those with the even first figure present **impossible** conditions that should **not** be satisfied if the syntagm is to be applied. Obviously, "odd" conditions are, implicitly, conditions with the existential quantifiers, requiring that there should exist at least one variable for which the condition is satisfied. Conversely, "even" conditions have implicit universal quantifiers: for every variable it should not be true that the condition is satisfied. Further, groups 1 and 2 of the CHECK zone list the conditions that could be checked using only morphological analysis results, the information from the dictionary entries of words present in the sentence processed and the linear order of the words in the sentence. Conditions belonging to groups with larger numbers can only be checked on the ready tree structure, or at least on the fragment thereof as it is generated by the parser. Once the conditions are satisfied, the instruction of the DO zone is performed. In sentence 5, the instruction establishes the hypothetical predicative link ("предик") going from X to Y.

1.2.2. An example of a dictionary entry

Fig. 2 below reproduces a simple combinatorial dictionary entry for the word *продажа* 'sale'. This is in fact only a part of the entry, from which the zone responsible for translation of the word into English is omitted (with the exception of the default translation field in line 24).

```
1   ПРОДАЖА
2         POR:S
3         SYNT:ЖЕНСК,ИСЧИСЛ
4         DES:'ДЕЙСТВИЕ','ФАКТ','АБСТРАКТ'
5         D1.1:ТВОР,'ЛИЦО'
6         D2.1:РОД
7         D3.1:ДАТ,'ЛИЦО'
8         D4.1:ЗА1,'ДЕНЬГИ'
9         D4.2:ПО4,НПУСТ,'ДЕНЬГИ'
10        _V0:ПРОДАВАТЬ
11        _SYN1:ТОРГОВЛЯ
12        _CONV:ПОКУПКА
13        _ANTI:ПОКУПКА
14        _S1:ПРОДАВЕЦ
15        _S2:ТОВАР
16        _S3:ПОКУПАТЕЛЬ
17        _OPER1:ОСУЩЕСТВЛЯТЬ
18        _OPER2:БЫТЬ<B2>
19        _INCEROPER2:ПОСТУПАТЬ1<B1>
20  TRAF:АГЕНТ.10
```

```
21  TRAF:1-КОМПЛ.20
22  TRAF:2-КОМПЛ.21
    ***************************
23  ZONE:EN
24  TRANS:SALE
…
```

Fig. 2. A lexical entry of the Russian combinatorial dictionary

Lines 1–2 indicate the lemma and the part of speech (noun).

Line 3 cites two simple syntactic features that point to the feminine gender of *продажа* and the fact that it is a count noun. These features are used whenever grammatical agreement of the word is to be checked, or verify whether it may form a quantificative noun phrase. As a matter of fact, the notion of syntactic feature is the most important in ETAP; the system involves over 200 syntactic features, some of them very sophisticated, which determine whether or not the word can be part of a particular syntactic construction.

Line 3 presents semantic features, or descriptors, of the word: in this case 'action', 'fact', and 'abstract'. Descriptors are used to ensure semantic agreement between elements of the sentence processed. Unlike semantic features, the system of descriptors in ETAP is rather simple and straightforward: it includes ca. 40 elements arranged into a weak hierarchy.

Lines 5–9 provide the government pattern of the word.

Lines 10 to 19 list values of the different lexical functions (LF) for which *продажа* is the keyword. Of these, lines 10–15 introduce substitute LFs, and lines 17–19 list collocate LFs.

2. Essentials of the algorithm

2.1. Morphological analysis as input of ETAP algorithm

During text analysis, the parser proper operates after the morphological analyzer produced a morphological structure (MorphS) for each sentence. MorphS is the ordered sequence of all words of the sentence, each one represented by a lemma name, a POS attribute and a set of morphological features. If a word form is lexically and/or morphologically ambiguous, it appears in the MorphS as a set of objects, somewhat loosely called homonyms, each consisting again of a lemma name, a POS attribute and a set of morphological features.

The morphological analyzer is based on a comprehensive morphological dictionary of Russian that counts over 130,000 entries. ETAP has no separate POS tagger; however, there is a small post-morphological module that partially resolves lexical and morphological ambiguity taking account of near linear context. On average, the module purges less than 20 % of homonyms.

2.2. Creation of the Set of Hypothetical Syntactic Links

ETAP takes a MorphS of a sentence processed as **input** and builds a dependency tree for this sentence using syntagms. At the first stage of the algorithm, the parser constructs all possible hypothetical links, which is performed in a number of steps. The primary list (the so-called matrix of hypotheses) is built exclusively on account of the linear conditions of the syntagms (see above). After that, conditions belonging to groups with higher numbers are applied to the matrix: at this stage, different methods of backtracking are used.

After all conditions of the syntagms have been verified, the algorithm resorts to a number of **filters** aimed at deleting excessive links so that the remaining ones form a dependency tree.

These filters are of diverse nature and may involve

- data on agreement or government,
- repeatability/non-repeatability of specific syntactic relations (e. g. a verb may have several adverbial modifiers attached by the adverbial relation but only one subject or one direct object),
- data on link projectivity (by default, any link is projective unless a set of specific conditions are met).

Importantly, the parser has three sets of rules in addition to syntagms, resorted to in the process of tree generation. These include intersyntactic rules; top node selection rules and preference rules.

2.3. Intersyntactic rules

Intersyntactic (INTERSYNT) rules operate on the whole set of hypothetical rules after all conditions of syntagms have been checked. These rules are designed to **prioritize** the hypotheses produced so that the subsequent stages of the algorithm could first choose the hypotheses with higher priority. The rules assign certain **weights** to syntactic hypotheses as well as to different homonyms of an ambiguous word on the basis of empirically found regularities that involve POS information, type of lexical ambiguity, certain syntactic configurations and the like. At present, this is only done by instructions that increase or reduce the strength of a link or a homonym and do not resort to any numerical values. Accordingly, the newly assigned weights are **absolute** (i. e. we cannot reduce or increase the weight of one link or homonym with respect to another concrete link or homonym). Despite this, INTERSYNT shows a rather satisfactory performance, which positively affect the quality of the parser.

Some of the INTERSYNT rules take account of **lexical co-occurrences**. E.g. if a sentence contains a collocation that is likely to be considered as an argument of a lexical function and its value, such a collocation is prioritized: the link that connects the part of such a collocation and/or homonyms that constitute it are assigned high weight values.

An important recent innovation in this mechanism is the creation of INTERSYNT rules that in fact reproduce the most important syntagms, which form the bulk of the syntax, with a vital difference that the syntagms' conditions are formulated for a drastically simplified environment (shorter distances between the head and the daughter, default word order ignoring rarely occurring inversions, prototypical instantiations of variables, e. g. only nouns are included but not their syntactic equivalents like numerals. substantivized adjectives or participles etc.). If in a sentence processed the conditions of such a rule are met, the respected link is assigned a high weight value; respectively, the link generated by the "parent" syntagm is likely to appear in the resulting tree (Tsinman-Druzhkin 2008).

2.4. Top Node selection rules

The so-called **top node selection** rules arrange possible candidates for the absolute head of the future tree structure according to the empirical likelihood principle. Handwritten rules of this ordering take account of a number of different factors (part of speech, morphological features, linear position in the sentence, close environment, presence or absence of hypothetical links going to and from the word tested etc.) and perform fairly well. For example, a finite verb X_1 is more likely to act as head of the sentence than another finite verb X_2 located to the right of X_2; however, the situation reverts if X_1 is preceded by a subordinating conjunction, in which case X_1 will probably depend on this conjunction in the subordinate clause and X_2 will be more likely to act as absolute head.

This block of rules is the only one in ETAP when weight values could be relative (there are rules that increase or decrease the weight of some link with regard to another link, whose weight has been established previously).

A recent innovation in this block of rules is the inclusion of **a statistical component**: statistical data are collected from SynTagRus (see below).

2.5. General Preference Rules

Preference rules of several types are applied after all intersyntactic rules have been applied and the head is selected. The objective of preference rules is the same as that of INTERSYNT rules: prioritization of the remaining hypotheses. However, preference rules, unlike INTERSYNT rules, are not irreversible and the algorithm may roll back if at a particular step the construction of the structure is blocked.

Most preference rules work with syntactic hypotheses, trying to determine which one of a bunch of hypothetical links going to or from a particular word is the most plausible. Other rules prioritize different homonyms of words.

If after all these rules have been applied and no tree can be chosen because extra hypotheses are still present, the algorithm resorts to the exhaustion of the remaining alternatives. In the standard situation, the algorithm starts by eliminating one link of the remaining set, finding it in accordance with the preset graph transversal subroutine, uses recursion and rollback mechanisms if needed, until a tree is produced.

Recently, a modified technique of alternatives exhaustion has been introduced, which once again resorts to statistics collected from SynTagRus. This technique uses a greedy algorithm of choosing the links to be deleted based on the evaluation of probabilites of their correctness. To collect evaluation data, the parser is run on SynTagRus sentences, in which for every pair of alternating hypotheses we know which of them is correct, or know that both are incorrect. Pairwise probabilities are used to assess the correctness probabilities of for every link belonging to bunches of links entering a word. The evaluation only taken account of names and lenghths of the links and is therefore rather rough but has proven to be fairly efficient.

2.6. Patterns of ETAP operation

ETAP parser has three patterns of operation, which are called **rapid syntax**, **full syntax**, and **emergency syntax.** The first pattern may be started after INTERSYNT rules have been applied: the algorithm temporarily deletes all weak links and homonyms and strives to build the tree from strong and normal elements alone. If this pattern fails, the algorithm restores the weak links and resumes the work with the whole set of hypothesis: this is the full syntax pattern. Should this pattern fail, too, the algorithm resorts to the emergency syntax pattern, which starts by detecting the node or nodes left without the head and attaching it to some other nodes with the help of a fictitious syntactic link or links, using the so-called soft-fail mechanism. If emergency syntax is activated, the resulting tree may prove to be far from satisfactory.

ETAP options allow the user to skip either the rapid syntax or the full syntax pattern, but not both.

3. Major applications

3.1. ETAP-3 machine translation system

Originally, ETAP parser of Russian was intended for machine translation and built specifically for this purpose. Together with the parser for English it constituted the main computational linguistics resource on which the system is based.

This objective naturally determined many of the properties of the parser and concrete solutions taken therein; in particular, the developers placed a very strong emphasis on the lexical aspect of the system, primarily striving to represent in the most precise manner all links that were responsible for the instantiation of valencies of the predicates, while the achievement of overall syntactic accuracy was given a somewhat lesser priority. In some cases, decisions were taken to deliberately disregard certain linguistic phenomena in order to simplify the rules. For example, the parser does not build non-projective attributive and adverbial links, although actant links like predicative and 1st completive may well be non-projective.

3.2. SynTagRus treebank of Russian

Another important application of ETAP is the creation of the first syntactically tagged corpus of Russian, SynTagRus (see e. g. Apresjan et al. 2005)[6]. The corpus is built semiautomatically: for every sentence of a text belonging to the corpus ETAP first builds a syntactic tree, which is then manually checked by at least two human experts, which ensures high quality of the corpus. Human work is facilitated by a powerful software environment, called Structure Editor, which provides a variety of aids to make the process of corpus editing effective and minimize the number of errors (Iomdin-Sizov 2009). It may happen that ETAP cannot at all build a syntactic tree for the sentence (e. g. if it contains an ellipsis); in this case the expert constructs the tree manually, introducing phantom nodes as needed.

At present, the corpus counts a little over 50,000 sentences (over 460,000 words). Despite this relatively limited size, the corpus proves to be extremely useful not only as a linguistic resource but also as a computational resource which can be utilized to collect various statistical data, create training sets for machine learning, and develop automatic parsers (see Nivre-Bogusalvsky-Iomdin 2008). One of the new features of SynTagRus is that it provides, in addition to syntactic annotation, also annotation with collocate lexical functions.

Importantly, SynTagRus is now effectively used by the ETAP parser itself. There are three main uses of the corpus.

First, it provides the statistics of occurrence of the different syntactic constructions, lexical co-occurrences, patterns of ambiguities etc., which is used in several points of the algorithm if the statistical component is activated.

Second, it serves as an efficient and rather accurate evaluation resource, which is used to evaluate the performance of ETAP parser in many respects and so find and resolve some of the system's bottlenecks (see Boguslavsky et al. 2011).

Finally, it is used for regression testing of ETAP. Periodically, ETAP is run on the whole material of the corpus. Sentences that receive parses exactly equivalent to those stored in the corpus (this subset constitutes between 30 and 35 percent of the bulk of the corpus) are selected as basis for regression testing. ETAP is then regularly run on this test set to see if any of the changes introduced in the dictionary, rules, or software mechanisms affected the state of the test set. Regression testing has proven extremely helpful in ensuring the stability of the parser and eventually improving it in many respects.

3.3. A hybrid system of Russian speech synthesis

ETAP parser has been effectively used in creating a new system of Russian speech synthesis, ETAP-Multiphone (see Iomdin-Lobanov 2009, Iomdin-Lobanov-Getsevich 2011). The idea is that prior to sending the text to the regular synthetic block it is parsed by ETAP supplemented with rules that find prosodically salient elements

[6] SynTagRus is accessible online on the website of the Russian National Corpus (www.ruscorpora.ru) as its subcorpus.

in the syntactic structure. The elements receive special treatment in the regular synthetic block, which noticeably improves the result of speech generation. Within this project, the morphological dictionary of ETAP was supplemented with information on the phonetic stress of every word form, which naturally included correct rendering of the Russian letter *ë*. This helped improve the performance of ETAP in sentences where words that may be written with *ë* are indeed written in this way.

3.4. A semantic analyzer of text involving an ontology

ETAP parser is used in all new systems that are based on, or constitute a part of, the ETAP-3 linguistic processor. One such system is the semantic analyzer of Russian texts that makes use of a specially designed ontology (see Boguslavsky et al. 2010). The new system requires that the parser performs as accurately as possible. Among other things, the parser must ensure that arguments and values of lexical functions occurring in the text processed could be identified correctly. This provides additional incentives for ETAP development.

4. Unsolved problems and future development

To conclude the description of ETAP we will briefly outline the challenges that the system is still facing. The most important challenge is that, so far, the system is not sufficiently robust. In certain cases, the parser fails to produce an adequate or even an acceptable tree structure. This maybe due to a variety of reasons.

The first reason is that the system cannot work reliably on very long sentences (60 words or more) due to the combinatorial explosion and the fact that it has no good heuristic mechanisms of splitting such sentences into linguistically acceptable chunks.

The second reason is that ETAP lacks sufficient external resources, like a named entity recognition component, POS tagger, or a reliable morphological guesser, which reduces its potential of correctly handling sentences with unknown words.

In some cases, linguistic support of ETAP has obvious gaps. In particular, this is manifested in the fact that linguistic rules are sometimes too rigid and are unable to cope with sentences that contain deviations of the prescribed standard (metaphorical uses of words, irregular instantiation of valencies and the like); besides, it has no proper mechanisms of handling elliptical sentences of many kinds.

Additionally, ETAP has certain inadequacies in the core algorithm. In particular, soft-fail mechanisms that are used in the emergency syntax pattern of operation are rather rough and, instead of providing a structure with only local defects, may sometimes play havoc with the result.

All these challenges are now being addressed. The developers of ETAP are working to create the necessary resources, including the POS tagger and the morphological guesser, partially using machine learning techniques. Special efforts are also made to convert the parser into a hybrid system that combines rule-based and machine learning approaches.

References

1. *Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin* et al. (1989). Lingvisticheskoe obespechenie sistemy ETAP-2 [The linguistics of the ETAP-2 MT system]. Moscow, Nauka. 295 p.
2. *Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin* et al. (1992). Lingvisticheskij protsessor dlja slozhnyx informatsionnyx sistem [A linguistic processor for advanced information systems]. Moscow, Nauka, 1992. 256 p.
3. *Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin, Alexandre Lazourski, Vladimir Sannikov, Victor Sizov, Leonid Tsinman.* (2003). ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT, MTT 2003, First International Conference on Meaning — Text Theory (June 16–18 2003). Paris: Ecole Normale Supérieure, P. 279–288.
4. *Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin, Vladimir Sannikov* (2010). Teoreticheskie problemy russkogo sintaksisa. Vzaimodejstvie grammatiki I slovarja [Theoretical Issues of Russian Syntax. Interaction of the Grammar and the Dictionary]. Moscow. Yazyki Slavyanskix kultur publishers. ISBN 978-5-9551-0386-0. 408 p.
5. *Jurij Apresjan, Leonid Iomdin* (1990). Konstruktsii tipa NEGDE SPAT' v russkom jazyke: sintaksis i semantika [Constructions of the NEGDE SPAT' type in Russian: Syntax and semantics.], Semiotika i informatika [Semiotics and Informatics], No. 29. Moskva, 1990, pp. 3–89.
6. *Jurij Apresian, Leonid Iomdin, Boris Iomdin* et al. (2005). Sintaksicheski i semanticheski annotirovannyj korpus russkogo jazyka (sovremennoe sostojanie I perspektivy [Syntactically and Semantically Annotated Corpus of Russian: State-of-the-Art and Prospects] In Natsionalnyj korpus russkogo jazyka 2003–2005 g. (rezul'taty i perspektivy). [National Corpus of Russian 2003–2005 (Results and Prospects)]. Moscow, Indrik. P.193–214.
7. *Igor Boguslavsky, Leonid Iomdin, Victor Sizov, Svetlana Timoshenko* (2010). Interfacing the Lexicon and the Ontology in a Semantic Analyzer, COLING 2010. Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010). Beijing, August 2010. P. 67–76.
8. *Igor Boguslavsky, Leonid Iomdin, Leonid Tsinman, Victor Sizov, and Vadim Petrochenkov* (2011). Rule-Based Dependency Parser Refined by Empirical and Corpus Statistics, International Conference on Dependency Linguistics. exploring dependency grammar, semantics, and the lexicon. Kim Gerdes, Eva Hajicova, Leo Wanner (eds). Depling 2011, Barcelona, September 5–7 2011. ISBN 978-84-615-1834-0. P. 318–327. http://depling.org/proceedingsDepling2011.
9. *Igor Boguslavsky, Leonid Iomdin, Victor Sizov, Denis Valeev.* (2008). Sintaksicheskij analizator sistemy ETAP i ego otsenka s pomoshchju gluboko razmechennogo korpusa russkix tekstov. [The syntactic analyzer of the ETAP system and its evaluation with the help of a deeply annotated corpus of Russian texts]. Труды Международной конференции "Корпусная лингвистика -2008" [Corpus Linguistic — 2008. International Conference]. Saint Petersburg, Saint Petersburg State University. ISBN 978-5-288-04769-5. p. 56–74.

10. *Leonid Iomdin, Boris Lobanov* (2009). Sintaksicheskie korreljaty prosodicheski markirovannyh elementov predlozhenija [Syntactic Correlates of Prosodically Marked Sentence Elements], Dialog 2009. Kompjuternaja lingvistika i intellektual'nye texnologii. Po materialam ezhegodnoj Mezhdunarodnoj konferentsii "Dialog" (Bekasovo, 27–31 maja 2009 g.). [Dialog 2009. Computational Linguistics and Intellectual Technologies. International Conference]. Moscow, RGGU Publishers, 2009. Issue 8(15). ISBN 978-5-7281-1102-3. P. 136–142.
11. *Leonid Iomdin, Boris Lobanov, Yuri Getsevich* (2011). Govorjashchij ETAP. Opyt ispol'zovanija sintaksicheskogo analizatora sistemy ETAP v russkom rechevom sinteze. [Talking ETAP. Using the Syntactic Analyzer of the ETAP System in Russian Speech Synthesis], Kompjuternaja lingvistika i intellektual'nye texnologii. Po materialam ezhegodnoja Mezhdunarodnoj konferentsii "Dialog" (Bekasovo, 25–29 maja 2011 g.) [Dialog 2011. Computational Linguistics and Intellectual Technologies. International Conference]. Moscow, RGGU Publishers, Issue 10(17). ISSN 2221-7932. P. 269–279.
12. *Leonid Iomdin, Victor Sizov* (2009). Structure Editor: a Powerful Environment for Tagged Corpora, MONDILEX Fifth Open Workshop, Ljubljana, Slovenia, 14–15 October, 2009. Ljubljana. ISBN 978-961-264-012-5. P. 1–12.
13. *Igor Mel'čuk* (1974/1999). Opyt teorii lingvisticheskih modelej Smysl ⇔ Tekst. [The theory of linguistic models "Meaning ⇔ Text']. Moscow, Nauka; Jazyki russkoj kultury.
14. *Joakim Nivre, Igor Boguslavsky, Leonid Iomdin* (2008). Parsing the SYNTAGRUS Treebank of Russian, Coling 2008. 22nd International Conference on Computational Linguistics. Proceedings of the Conference. Vol. 2. ISBN: 978-1-905593-47-7. P. 641–648.
15. *Leonid Tsinman, Konstantin Druzhkin* (2008). Sintaksicheskij analizator lingvisticheskogo protsessora ETAP-3: Èksperimenty po ranzhirovaniju sintaksicheskih gipotez [The syntactic analyzer of the ETAP-3: experiments on prioritizing syntactic hypotheses], Dialog 2008. Kompjuternaja lingvistika i intellektual'nye texnologii. Po materialam ezhegodnoja Mezhdunarodnoj konferentsii "Dialog" (Bekasovo, 4–8 ijunja 2008 r. Computational Linguistics and Intellectual Technologies. International Conference]. Moscow, RGGU Publishers, Issue 7(14). P. 147–153. ISBN 978-5-7281-1022-4.

Abstracts

SYNTACTIC AND SEMANTIC PARSER BASED ON ABBYY COMPRENO LINGUISTIC TECHNOLOGIES

Anisimovich K. V. (Konstantin_An@abbyy.com), **Druzhkin K. Ju.**(Konstantin_D@abbyy.com), **Minlos F. R.** (f.minlos@gmail.com), **Petrova M. A.** (Maria_P@abbyy.com), **Selegey V. P.** (Vladimir_S@abbyy.com), **Zuev K. A.** (Konstantin_Z@abbyy.com), ABBYY, Moscow, Russia

The paper presents Abbyy Syntactic and Semantic Parser that was a participant of the Dialog 2012 Syntactic Parsers Testing Forum. We will refer to the parser technology (both parsing algorithms and linguistic model) as Compreno technology. We do not touch on any evaluation issues, as they are tackled by the Forum panel. Instead, the paper makes public some underlying principles of the parser. What we want to communicate directly concerning the testing are the features of the project which are both relevant to the comparison of our results with the "gold standard" adopted by the panel and, at the same time, important for the whole architecture of our technology.

RUSSIAN DEPENDENCY PARSER SYNTAUTOM AT THE DIALOGUE-2012 PARSER EVALUATION TASK

Antonova A. A. (antonova@yandex-team.ru), **Misyurev A. V.** (misyurev@yandex-team.ru), Yandex

In this paper we describe the SyntAutom parser submitted at the Dialogue-2012 parser evaluation task. It is a rule-based system, which performs syntactic and morphological ambiguity resolution in a unified framework. The underlying grammar formalism is Dependency Grammar. The segmentation, shallow parsing and deep parsing are based on a special form of finite-state pushdown automatons, implemented as a sets of recursive functions. The input of the automaton is a lattice of objects — these may be words or shallow trees. Within the functions we have implemented a mechanism of branching and multiple return — a possibility for a function to generate a bunch of states. We discuss the system architecture, the output parsing trees structure and the common types of incorrect analyses. The distinctive feature of the system is that it tends to directly connect meaningful words, while auxiliary words are demoted to the lower tree levels. Although the system experiences the common problems of most rule-based systems, it has proven its applicability in a range of real-world applications.

TESTING THE SENTIMENT CLASSIFICATION APPROACH IN VARIOUS DOMAINS — ROMIP 2011

Chetviorkin I. I. (ilia2010@yandex.ru), Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University

We offer a review of sentiment classification experiments in various domains using different training sets. In the movie domain we studied the impact of opinion word weights on the quality of classification. We selected the best feature set and ran them on each task-domain pair. In several tasks our algorithm achieved high quality of the classification.

SENTIMENT ANALYSIS TRACK AT ROMIP 2011

Chetviorkin I. I. (ilia2010@yandex.ru), Lomonosov Moscow State University, **Braslavski P. I.** (pbraslavski@acm.org), Kontur Labs, Ural Federal University, **Loukachevitch N. V.** (louk_nat@mail.ru), Research Computing Center of Lomonosov, Moscow State University

Russian Information Retrieval Seminar (ROMIP) is a Russian TREC-like IR evaluation initiative. In 2011 ROMIP launched a new track on sentiment analysis. Within the track we prepared a training collection of user reviews along with ratings for movies, books, and digital cameras. Additionally, we compiled a test collection of blog posts with reviews in the same domains and labeled them

according to expressed sentiment. The paper describes the collections' characteristics, track tasks, the labeling process, and evaluation metrics. We summarize the participants' results and make suggestions for future editions of the track.

ETAP PARSER: STATE OF THE ART

Iomdin L. (iomdin@iitp.ru), **Petrochenkov V.** (petrochenkvov@iitp.ru), **Sizov V.** (sizov@iitp.ru), **Tsinman L.** (cinman@iitp.ru), Institute of Information Transmission Problems (Kharkevich Institute), Russian Academy of Sciences

The state of the art of the ETAP-3 syntactic parser, which took part in a recent competition of Russian parsers, is presented. The paper gives an outline of the main linguistic resources involved in the parser's operation, describes the main features and steps of the algorithm, and briefly discusses the applications in which the parser is used, including a machine translation system, a software environment for the creation of a syntactically tagged corpus of Russian, and a hybrid system of Russian speech synthesis. Special attention is given to concrete scientific approaches and solutions that determine the functioning of the parser, including methods of lexical and syntactic disambiguation.

SENTIMENT ANALYSIS OF TEXTS BASED ON MACHINE LEARNING METHODS

Kotelnikov E. V. (kotelnikov.ev@gmail.com), **Klekovkina M. V.** (klekovkina.mv@gmail.com), Vyatka State University of Humanities, Kirov, Russian Federation

We present the methods of text processing and machine learning used to fulfill the tasks of the tracks for the sentiment analysis on the seminar ROMIP-2011. The issues of the choice of the optimal variant of text vector model and the most suitable machine learning method are addressed. Unsupervised and supervised TF.IDF methods of text representation are used. We apply such classification methods as: Naive Bayes, Rocchio's method, k-Nearest Neighbors, Support Vector Machines (SVM), the method based on keywords and the method which combines SVM and the keywords method. The experiments proved that the best way of text representation is unsupervised binary model with cosine normalization. The combination of SVM and keywords method showed the best results for classification. The authors give the analysis of the results in comparison with other participants of ROMIP-2011.

LANGUAGE INDEPENDENT APPROACH TO SENTIMENT ANALYSIS (LIMSI PARTICIPATION IN ROMIP '11)

Pak A. (alexpak@limsi.fr), **Paroubek P.** (pap@limsi.fr), Universite Paris-Sud, France

Sentiment analysis is a challenging task for computational linguistics. It poses a difficult problem of identifying user opinion in a given text. In this paper, we describe participation of LIMSI in the sentiment analysis track of the Russian annual evaluation campaign (ROMIP'11). The goal of the track was classification of opinions expressed in blog posts into two, three, and five classes. Our system based on SVM with dependency graph and ngram features was placed 1st in 5-class task on all three datasets (movies, books, cameras), 3rd in the 2-class task on the movies dataset, and 4th in the 3-class task on the cameras dataset, according to the official results.

RESEARCH ON APPLICABILITY OF THEMATIC CLASSIFICATION METHODS TO THE PROBLEM OF BOOK REVIEW CLASSIFICATION

Polyakov P. Yu. (pavel@rco.ru), **Kalinina M. V.** (kalinina_m@rco.ru), **Pleshko V. V.** (volodia@rco.ru), RCO LLC, Moscow, Russian Federation

The paper examines the different approaches to forming the training set, methods for extracting classification features, as well as methods of constructing classifiers regarding the problem of book review sentiment analysis. The tasks were to divide book reviews into 2 groups (positive, negative) and into 3 groups (positive, negative, neutral). Several methods were tested in the solution of the two tasks. It was shown that good results could be obtained by using common document categorization methods. The obtained figures approach the best results of the Web-site and regulatory doc-

ument classification track achieved by participants of ROMIP seminar. A method for enrichment of classification features within the linguistic approach using evaluative vocabulary dictionaries was proposed. It was established that this method gives a slight improvement in the results for the binary classification. We plan to explore in more detail the possibility of using expert-linguistic approaches to the construction of classification features.

PROOF OF CONCEPT STATISTICAL SENTIMENT CLASSIFICATION AT ROMIP 2011

Poroshin V. (vladimir.poroshin@m-brain.com), M-Brain Oy, Helsinki, Finland

In this paper we present a simple statistical classification method that predicts whether the opinion expressed by text in natural language is positive or negative. There are two main approaches in the sentiment or opinion detection: linguistic rule based systems and statistical algorithms. While statistical methods are easier to build when sufficient training data is available, it is widely perceived that a linguistic system can deliver better results. Our work was intended to prove the concept that a simple Naive Bayes based statistical classification algorithm with a minor language dependent adaptation is able to perform well in a binary sentiment classification task. In order to prove the hypothesis, we participated in Russian Information Retrieval Seminar (ROMIP) 2011 sentiment classification track [1], and achieved quite competitive results in sentiment prediction of Russian blog posts. This paper contains a detailed description of our classification method, including a feature extraction and normalization process, training and test data, evaluation metrics; and presents our official ROMIP results.

NLP EVALUATION 2011–2012: RUSSIAN SYNTACTIC PARSERS

Toldova S. Ju. (toldova@yandex.ru), **Sokolova E. G.** (minegot@rambler.ru), Russian State University for the Humanities, Moscow, Russia, **Astaf'eva I.** (astafir@gmail.com), **Gareyshina A.** (a.r.gare@gmail.com), **Koroleva A.** (tresh_miralissa@mail.ru), **Privoznov D.** (dprivoznov@gmail.com), **Sidorova E.** (begushchaya.po.volnam@gmail.com), **Tupikina L.** (lyubov98@gmail.com), Lomonosov Moscow State University, Moscow, Russia, **Lyashevskaya O. N.** (olesar@gmail.com), National Research University Higher School of Economics, Moscow, Russia

NLP Evaluation forum RU–EVAL started in 2010 as a new initiative aimed at independent evaluation of the methods used in Russian language resources and linguistic tools. The second evaluation campaign (2011–2012) is focused on syntactic parsing. It is open both to academic institutions and industrial companies and its general objective is to access the current state–of–the–art in the field and promote the development of syntactic technologies. The paper presents the principles and design of two tracks, which were organized thematically, namely, Main track and News. There were seven participants who follow either rule–based or statistical approach; all of them submitted runs to both tracks. The training set consisted in 100 sentences, the dataset for annotation included ca. one million words, and the test set was composed by ca. 800 sentences (500+ sentences for the Main track and 300+ sentences for the News track). The test set was annotated manually as a Golden Standard by two annotators. We describe how the outputs were compared and discuss common pitfalls for evaluation as well as some cases that are still problematic for parsing. As a side effect of the evaluation campaign and benchmarks for future development, the test data including Golden Standard and three automatically annotated answers are available to the NLP community at http://testsynt.soiza.com.

SENTIMENT CLASSIFICATION BY FRAGMENT RULES

Vasilyev V. (vvg_2000@mail.ru), **Khudyakova M.** (mariya.kh@gmail.com), **Davydov S.** (davydov_sergey@hotmail.com), LAN-PROJECT, Moscow, Russia

In this paper approaches to sentiment classification based on using fragment rules are described. Rules are constructed manually by experts and automatically by using machine learning procedures. Training sets, evaluation metrics and experiments are used according to ROMIP 2011 sentiment analysis track.

Авторский указатель

Научное издание

Компьютерная лингвистика и интеллектуальные технологии

По материалам ежегодной
Международной конференции «Диалог»

Выпуск 11 (18). 2012

Том 2. Доклады специальных секций

Ответственный за выпуск **В. Л. Талис**
Вёрстка **К. А. Климентовский**

Подписано в печать 14.05.2012
Формат 152 × 235
Бумага офсетная
Тираж 200 экз. Заказ № 224

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