Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2016"

Moscow, June 1–4, 2016

ENTITY BASED SENTIMENT ANALYSIS USING SYNTAX PATTERNS AND CONVOLUTIONAL NEURAL NETWORK

Karpov I. A. (karpovilia@gmail.com)[1],
Kozhevnikov M. V. (kozhevnikov1511@gmail.com)[2],
Kazorin V. I. (zhelyazik@mail.ru)[2],
Nemov N. R. (nemo_1@pisem.net)[2]

[1]National Research University Higher School of Economics, Moscow, Russia

[2]Research and Development Institute «Kvant», Moscow, Russia

This paper provides an alternative method to extracting object-based sentiment in text messages, based on modified method previously proposed by Mingbo [8], in which we first parse the syntax, and then correlate the sentiment with the object of analysis (also referred to as entity by some, therefore, used in this article interchangeably). We show two approaches for the sentiment polarity classification: syntactic rule patterns and convolutional neural network (CNN). Even without domain specific vocabulary and sophisticated classification algorithms, rule-based approach demonstrates an average macro-F_1 based rank among the participants, whereas domain-specific vocabularies show a slightly higher macro-F_1 score, but still close to an average result. CNN approach uses syntax dependencies and linear word order to obtain more extensive information about object relations. Convolution patterns, designed in this approach, are very similar to rules, obtained with rule-based approach. In our proposed approach, the neural network was trained with different Word2Vec (WV) models; we compared their performance relative to each other. In this paper, we show that learning a domain-specific WV offers slight progress in performance. Resulting macro-F_1 score show performance in the into top three of the overall results among the competitors, participating in 2016 SentiRuEval event. Originally, we have not submitted our results to this competition at the time it was held, but had a chance to compare them post-hoc. We also combine the CNN approach with the rule-based approach and discuss the obtained differences in results. All training sets, evaluation metrics and experiments are used according to SentiRuEval 2016.

Keywords: sentiment analysis, object-oriented sentiment analysis, syntax patterns, machine learning, convolution neural network

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ ТОНАЛЬНОСТИ ПРИ ПОМОЩИ СИНТАКСИЧЕСКИХ ШАБЛОНОВ И СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Карпов И. А. (karpovilia@gmail.com)[1],
Кожевников М. В. (kozhevnikov1511@gmail.com)[2],
Казорин В. И. (zhelyazik@mail.ru)[2],
Немов Н. Р. (nemo_1@pisem.net)[2]

[1]НИУ ВШЭ, Москва, Россия

[2]НИИ «Квант», Москва, Россия

Ключевые слова: определение тональности, тональность объектов, синтаксические шаблоны, машинное обучение, сверточные нейронные сети

1. Introduction

The online reputation analysis task, performed on social networks' data such as Twitter data, has several differences from the traditional sentiment tasks. We suggest that performance of systems designed to solve this problem depends on three factors:

(i) **Lexicon actualization**—the first issue is that there are many texts that do not contain any intuitively subjective words, but nonetheless, express a person's attitude. Usually such words are domain-specific. For example, in the context of everyday media usage of the word, the verb "*выдавать*" (most closely translated as to "fib"), has negative sentiment, because it is frequently used in meaning "*to lie*" ("*представлять что-либо не тем, чем оно является на самом деле*") or "*to betray*" ("делать донос, предавать")[1]. However, in banking, the same word means "*to issue a credit card*" or "*provide a loan*" ("*передавать в чье-л. распоряжение*") and usually has a positive sentiment. A promising approach to sentiment word extraction was described in [1]. In this paper, we study different Word2Vec models, trained by using news and social networks data.

The second issue is that pejorative lexicon used by social network users does not always indicates a negative opinion. For example, someone may be using swear language to indicate either negative or positive affect, which may not be obvious immediately.

[1] Meanings of the verb "*выдавать*" are provided by WikiDictionary: http://ru.wiktionary.org/wiki/выдавать

(ii) **Object matching**—the issue here is linking the sentiment word with key object, especially when text is long or when there are multiple entities mentioned. For example, it would be very difficult to analyze the sentence "***Билайн**, которым я пользовался два года, гораздо **лучше МТС***" ("*Beeline, that I've used for two years, is much better than MTS*") using only linear context because key sentiment word "*лучше*" is much closer in absolute word distance to the object "*MTS,*" rather than "*Beeline*." Also, according to our experience, analysis of comparison structures such as "*A is better than B,*" without syntax information, produces erroneous results. Both our approaches incorporate syntax information, as described later in the paper. With that, we are basing our method on the classical approaches to solving this problem as described in [9], [11].

(iii) **Subjective fact interpretation**—recent sentiment evaluation competitions show tendency of adding fact interpretations to sentiment analysis. For example, in the sentence "***Сбербанк** подаст в суд иск по банкротству **Мечела***" ("***Sberbank** will bring a bankruptcy case against **Mechel** to court*"), we have a fact of a bankruptcy, negative for "*Mechel*", but an ordinary bank activity for "*Sberbank*." Processing such data requires many specific, often counteractive rules to deal with the problem of contradicting sentiments in the traditional rule-based approach, but could be efficiently performed by modern neural networks.

Recent works involving CNN-based approaches in English [8], [4], [2] have demonstrated excellent results on various classification tasks, including sentiment analysis. Because we expected that (ii) and (iii) factors could only be solved with syntax-dependency information, we used CNN, which uses not only linear word order, but also syntax dependencies to extract sentiment, and could allow for more efficiency in the task processing.

Rule-based approach, described later in this paper, is similar to the RCO approach [4], but there are differences in text preprocessing and lexical dictionaries' extraction. The CNN approach is also similar to [8] paper, but we have changed the input vector and made entity token with special TARGET mark to achieve a more efficient object-oriented sentiment analysis. We also used custom convolution patterns in this work.

2. Methods

Figure 1 gives a brief overview of the proposed approach. Input text is parsed with graphematic, morphological, and syntax parsers at the Text preprocessing stage. The rule-based approach assumes that predefined syntax patterns are enhanced by preliminarily generated Word2Vec models and sentiment dictionaries. as Resulting feature vector is analyzed by the extremely naive classifier that labels the object sentiment according to quantity of sentiment facts, linked with this object in the text. The resulting sentiment is a net sum of positive and negative sentiment labels. In case of the CNN-based approach, preprocessed text is vectorized with preliminary generated Word2Vec model. CNN returns the sentiment label as a result. We first build two separate classifiers, which can be easily combined, as shown in experiments section later in the paper. We now discuss each module in detail.

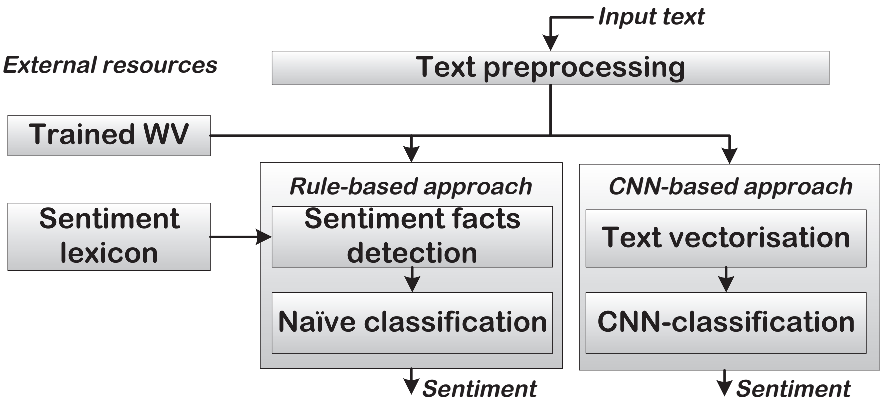


Fig. 1. Overall system architecture

2.1. Text preprocessing

Since input data from social networks is very noisy, a substantial amount of preprocessing is required. These steps are discussed below.

2.1.1. Remove URLs

URLs do not carry a lot of substantial information regarding the sentiment of the tweet and contaminate the dictionary, so we remove them with simple regex.

2.1.2. Remove nontextual data

Hashtags and tokens, starting with an “at” sign (@) represent important information about the reviewed object. In order to find it, we remove certain punctuation such as quotation marks, hyphens, asterisks, “at” signs, etc.

2.1.3. Tokenisation & morphology

We applied our own NLP toolkit [3] and Mystem parser developed by Yandex[2] for text preprocessing. Morphological analysis shows similar results, but tokenization, done by Mystem, was not designed to handle emotions and other punctuation specifics of social networks, so we preferred our own parser, which could overcome these limitations.

2.1.4. Named Entity (NE) recognition

We used Wikipedia hyperlink structure to find entities and their possible occurrences in the text as proposed in [12]. The basic algorithm was enhanced by adding transcripts and translations for each separately occurring appearances of key objects. We also generate separate grammatical cases for each normal form of the word

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

or phrase, describing the key object, and add them as a possible occurrence of key object in the text. As a result, we formulate the dictionary of the key objects' occurrences in the text. During the text processing step we replace key objects' occurrence with a special "TARGET" token and an appropriate morphology information.

2.1.5. Syntax parsing

We process entire dataset with malt parser [10], trained on our own news corpora to get dependency trees used by both approaches. If the tweet contains multiple root nodes, they are all added as descendants of special fake "*ROOT*" node. Sample syntax parse result is shown at figure 2. In general, our constructs had a single root, but in case it was not so, we used the described approach.

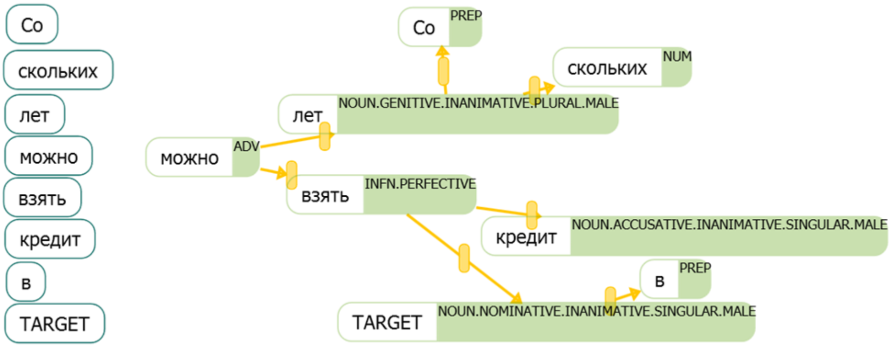

Со
скольких
лет
можно
взять
кредит
в
TARGET
Со
PREP
скольких
NUM
лет
NOUN.GENITIVE.INANIMATIVE.PLURAL.MALE
можно
ADV
взять
INFN.PERFECTIVE
кредит
NOUN.ACCUSATIVE.INANIMATIVE.SINGULAR.MALE
в
PREP
TARGET
NOUN.NOMINATIVE.INANIMATIVE.SINGULAR.MALE

Fig. 2. Syntax parse result example

2.2. Word2Vec training

We use the Word2Vec (VW) [7] modeling in both the rule- and the CNN- based approaches. In case of a rule-based approach, WV is used for computing semantic similarity between sentiment words. In CNN, WV is needed to represent text as a matrix for the neural network input. WV is trained on word lemmas with part-of-speech codes. We exclude punctuation, conjunctions, prepositions, particles and short (less than 3 symbols) English words from the training data. We use 300-dimension vector size skip-gram model with the minimum cut-off for the number of words = 3 in all cases.

Corpora lexicon plays an important role in generating WV model. We gathered nearly 1.5 million twitter search results about general topics such as music, cinema, travelling, literature, sports, etc[3]. Obtained model takes into account the specifics of twitter language, but still suffers from the word sense ambiguity problem. Therefore, we also gathered twitter search results for banking and telecom topics of nearly 100,000 tweets each.

[3] Selected categories list, trained models and project code can be found at http://github.com/lab533/RuSentiEval2016

The following combinations of gathered corpora were made to find the balance between corpora size and word ambiguity problem:

- WV_Banks_clear: 120,000 bank tweets
- WV_TTK_ clear: 120,000 telecom tweets
- WV_Twitter: 1,500,000 gathered twits
- WV_news: 4,500,000 news texts

We also added news-based WV to explore the role of twitter-specific vocabulary in sentiment tasks. Different mixtures of gathered corpora was evaluated as described in experiments section.

2.3. Rule-based approach

As a first step, we look for sentiment words of a tweet. We use our own universal dictionary of sentiment words for this purpose. Dictionary consists of 2,074 positive and 6136 negative normal word forms, manually verified by experts. After inflection of normal words forms and their enrichment with top 2 most similar WV words, dictionary was transformed to 60,288 positive and 189,953 negative word forms. Using the syntax tree of the sentence, which contains sentiment word, we detect modal verbs and negotiation markers (like “*не*”, “*нет*” etc.).

Next, we define sentiment facts associated with sentiment words. Sentiment fact is a semantically isolated part of a syntactic tree, which contains the sentiment word. In our rule-based approach, there are two types of sentiment facts, depending on parent of the sentiment word. If a parent of the sentiment word is a subordinate part of a sentence, a sentiment fact is a branch of the syntax tree with the parent of the sentiment word. This is the first type of sentiment facts. An example of such fact is the phrase “*уродливое здание Сбербанка*” (“*ugly Sberbank building*”) of the sentence “*В каждом городе России есть уродливое здание Сбербанка*” (“*There is an ugly Sberbank building in each city in Russia*”), as shown at figure 3.

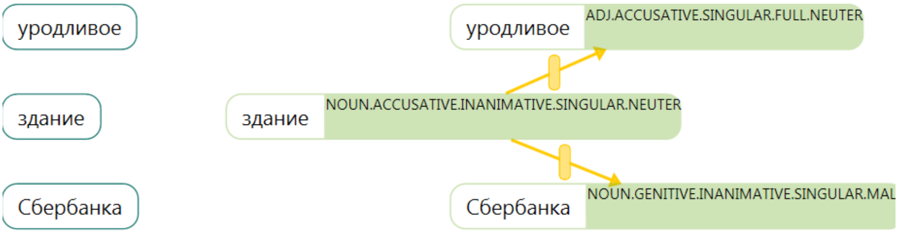

уродливое
здание
Сбербанка
уродливое
ADJ.ACCUSATIVE.SINGULAR.FULL.NEUTER
здание
NOUN.ACCUSATIVE.INANIMATIVE.SINGULAR.NEUTER
Сбербанка
NOUN.GENITIVE.INANIMATIVE.SINGULAR.MAL

Fig. 3. Sentiment fact with an adjective modifier sentiment word

The second type of a sentiment fact is the sentiment word or its parent, which is one of the subjects of the sentence or one of its predicates. In this case, the sentiment fact includes a predicate, a subject, and all of their children tokens. For example, the sentiment fact here is the "*ненавижу Райффайзен банк*" ("*hate Raiffaizen bank*") in the sentence "*Я не устану повторять, что ненавижу Райффайзен банк*" ("*I will never stop saying that I hate Raiffaizen bank*"), as shown at figure 4.

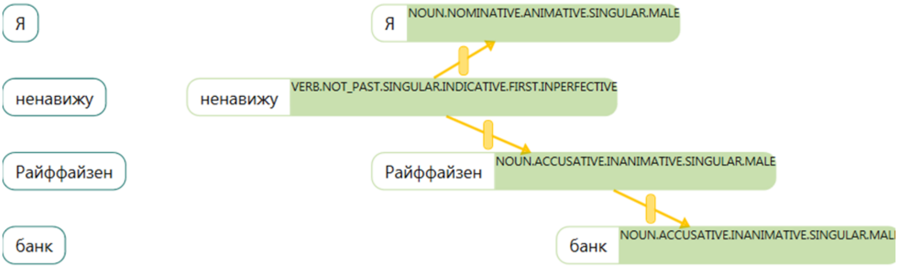

Я
ненавижу
Райффайзен
банк
Я
NOUN.NOMINATIVE.ANIMATIVE.SINGULAR.MALE
ненавижу
VERB.NOT_PAST.SINGULAR.INDICATIVE.FIRST.INPERFECTIVE
Райффайзен
NOUN.ACCUSATIVE.INANIMATIVE.SINGULAR.MALE
банк
NOUN.ACCUSATIVE.INANIMATIVE.SINGULAR.MALI

Fig. 4. Sentiment fact with a predicate sentiment word

Next, we unite neighboring sentiment facts: if one of the tokens of the sentiment fact has a syntactic connection with a token of another fact, these two facts get combined into one. Then we apply rules of combination of positive and negative sentiment words inside facts, and calculate integer sentiment index for each fact.

To improve general performance of the algorithm, we also made some individual rules for each domain:

- Stop-words list (words from dictionary that do not have any sentiment for a specific domain);
- Unigram and n-gram words list (words that have a sentiment value only for a specific domain);
- Applying "No-rule" (words or n-grams that have sentiment only with or without negotiation);

Finally, we find sentiment facts that contain a target object. If there is no sentiment fact with a target object, we assign object to the nearest fact in the syntactic tree. Then we calculate total sentiment score for each object and use it as a final sentiment result. We mark tweets that do not have any sentiment facts as neutral.

2.4. Convolutional neural network approach

Convolutional neural networks (CNNs), originally invented in computer vision [5], in recent years have been applied in many natural language processing (NLP) tasks such as authorship detection, question answering, and sentiment analysis. Let $x_i \in R^k$ be the k-dimensional word vector corresponding to the i-th word in the sentence. The sentence of length n can be described as

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \quad (1)$$

where $\oplus$ is the concatenation operator. Such vector is considered to be CNN input. A convolution operation involves a filter $w \in R^{hk}$, which is applied to a window of h words to produce a new feature. This filter is applied to each possible window of words in the sentence to produce a feature map. Max-overtime pooling [4] operation over the feature map is applied to capture the most important feature—one with the highest value—for each feature map. These features maps form the penultimate layer and are passed to a fully connected softmax layer, whose output is the probability distribution over labels.

2.4.1. Dependency-based Convolution

We are using the Mingbo's [8] approach to include syntax information into the classification process, where dependency-based convolution is described as follows:

$$x_{1:n} = x_i \oplus x_{p(i)} \oplus \ldots \oplus x_{p^{n-1}(i)} \tag{2}$$

where $p^n(i)$ returns the i-th word's n-th parent, which is recursively defined as:

$$p^n(i) = \begin{cases} p\left(p^{n-1}(i)\right) & if\ n > 0 \\ \quad i & if\ n = 0 \end{cases} \tag{3}$$

Text preprocessing notation and the peculiarities of twitter text often cause the TARGET node to be separated from the sentiment fact into a different sentence. In order to capture these long-distance dependencies in the entire tweet, we use sibling convolutions defined as:

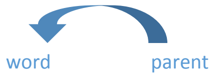
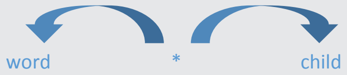
$$s(i,j) = \begin{cases} 1 & if\ p(i) = p(j) \\ 0 & if\ p(i) \neq p(j) \end{cases} \tag{4}$$

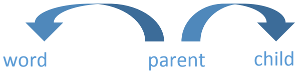
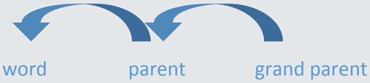
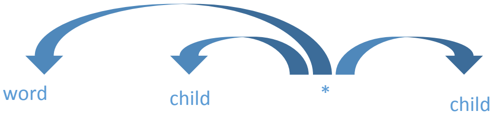
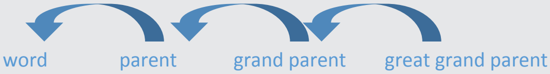
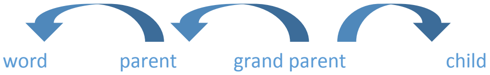
where $i > j$. We take maximum five first left siblings of i-th token to avoid combinatorial explosion.

2.4.2. Convolution patterns

Inspired by rule-based approach, we added several convolution patterns of length two to four words. Maximum pattern length was taken from the rule-based approach, where we have very few patterns longer than four tokens deep. It should be mentioned that one token doesn't equal one word, because we replace phrases with TARGET mark during object matching phase.

Table 1. Tree convolution patterns of different depth

Pattern depth	Pattern
2	
2	

Pattern depth	Pattern
3	
3	
3	
4	
4	

Asterisk in table 1 means that information about this word is not included to a convolution pattern. We also add information about the sequential token order in the tweet to compensate for parsing errors during the syntax analysis stage. The final input vector is a concatenation of feature maps from tree-based information and n-grams, with n=5.

2.4.3. Training

We substitute all "word + POS" pairs are by unique ids and align all sentences to length 50 (zero padding). We take first 5 anchestors and first 5 siblings for each word in a sentence and concatenate all words to form input vector for our NN. Neural network consists of the following layers:

- embedding layer—to turn word ids to word vectores, we used only words, contained in training;
- convolution layer—layer with rectified linear unit (ReLU) activation where convolution patterns are applied as described in table 1;
- maxPooling layer—which is down-sampling convolution layer output;
- dropout layer—with dropout rate was set to 0.25;
- dense layer—with ReLU activation;
- dropout layer—with dropout rate was set to 0.5;
- softmax layer—to form classification output.

We employ random dropout on penultimate layer to avoid overtraining as described in [4]. We trained our CNN for 40 epochs, but did not observe any increase in quality after the 2th epoch. Training was done through stochastic gradient descent over shuffled mini-batches with the AdaGrad update rule. Trained CNN models with exact parameters could be found at project repository, noted at section 2.2.

3. Experiments

Results of our evaluation are presented in Table 2. Consistent with standards of the RusSentiEval, the macro-averaged F_1-measurewas used as a primary evaluation metric [6]. Table 2 below describes positive and negative sentiment classes and micro-averaged F_1.

Table 2. Performance of rule- and CNN-based approaches in different configuration

Domain	Approach	Training collection	WV	F_1 positive	F_1 negative	Macro-average F_1	Micro-average F_1
Banks	Rule-based	Banks	—	0.387	0.501	0.443	0.463
	Rule-based with domain rules	Banks	—	0.394	0.524	0.459	0.482
	CNN	Banks	Random	0.425	0.555	0.490	0.523
		Banks	News	0.422	0.555	0.489	0.523
		Banks	Twitter	0.429	0.552	0.490	0.522
		Banks & TTK	Random	0.446	0.618	0.532	0.574
		Banks & TTK	News	0.455	0.611	0.533	0.572
		Banks & TTK	Twitter	0.456	0.615	**0.536**	0.574
Telecom	Rule-based	TTK	—	0.280	0.682	0.481	0.569
	Rule-based with domain rules	TTK	—	0.285	0.695	0.490	0.582
	CNN	TTK	Random	0.097	0.556	0.326	0.497
		TTK	News	0.091	0.557	0.324	0.499
		TTK	Twitter	0.091	0.559	0.325	0.500
		Banks & TTK	Random	0.307	0.738	0.523	0.681
		Banks & TTK	News	0.298	0.740	0.519	0.682
		Banks & TTK	Twitter	0.313	0.739	**0.526**	0.682

In the table above, the column "Training collection" describes the collection, chosen to train the model. In case of "Banks & TTK" value, model was trained on both Banks and Telecom data shuffled in random order. "WV" column describes Word2Vec model, used in the experiment. Results in Table 2 demonstrate that training corpora size is more important than the selected VW model. It also appears that WV is extremely sensitive to the input data. In our case VW, trained with only the domain specific data, shows better results that can be increased by acquiring bigger corpora.

3.1. Overall Performance

The evaluation metric used in the SentiRuEval 2016 competition is the macro-averaged F_1 measure calculated over the positive and negative classes. Table **3** shows the overall performance of our system for bank and telecom datasets.

Table 3. Performance of our method and best F_1 measure among all participants

Domain	Approach	F_1 positive	F_1 negative	Macro-average F_1	Micro-average F_1
Banks	Rule-based	0.394	0.524	0.459	0.482
	CNN	0.456	0.615	0.536	0.574
	Hybrid	0.457	0.619	**0.538**	**0.577**
	SentiRuEval best			0.552	
Telecom	Rule-based	0.285	0.695	0.490	0.582
	CNN	0.313	0.739	0.526	0.682
	Hybrid	0.313	0.740	**0.527**	**0.684**
	SentiRuEval best			**0.559**	

In case of rule-based approach, the system was not developed for banks or telecom companies' domains specially. Rule-based approach did not use any machine learning. Training collection was used only for extracting the proposed domain-specific rules, which approximately increased macro-average F-measure by 0.015.

With the Hybrid approach, final sentiment marks of neutral tweets, gained from rule-based approach, are inputs for a CNN. In general, rules give more precise result, but fail in recall. This method shows small performance progress in case of telecom domain, but does not help in bank domain, which may be caused by overfitting when multiple rules interfere each other.

4. Conclusions

We presented results of sentiment analysis on Twitter by building two approaches based on hand-written syntactic rules and CNN. Rule-based linguistic method showed average performance result, which makes it useful when training collection is not available. Few hand-written rules with well-filtered dictionaries can give a little boost to the CNN output, but the system degrades as rules count increases. CNN show very high quality result that coincides with the best results of the competition, but this approach requires relatively large training collections. The same problem occurs in distributive semantics, applied in this work. Word2vec can extract deep semantic features between words if training corpora is large enough.

Acknowledgment

The article was prepared within the framework of a subsidy granted to the HSE by the Government of the Russian Federation for the implementation of the Global Competitiveness Program.

References

1. *Chetviorkin I., Loukachevitch N.* (2012), Extraction of Russian Sentiment Lexicon for Product Meta-Domain. Proceedings of the 26nd International Conference on Computational Linguistics (Colling-2012), Mumbai, pp. 593–610.
2. *Kalchbrenner N., Grefenstette E., Blunsom P.* (2014), A Convolutional Neural Network for Modelling Sentences, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014), Baltimore, pp. 655–665.
3. *Karpov I., Goroslavskiy A.* (2012) Application of BIRCH to text clustering. Proceedings of the 14th All-Russian Conference “Digital Libraries: Advanced Methods and Technologies, Digital Collections” (RCDL-2012), Pereslavl Zalesskii, pp. 102–105.
4. *Kim Y.* (2014), Convolutional neural networks for sentence classification, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, pp. 1746–1751.
5. *LeCun Y., Cortes C., Bottou L., Jackel L.* (1995), Comparison of Learning Algorithms for Handwriting Digit Recognition, International Conference on Artificial Neural Networks, Paris, pp. 53–60.
6. *Loukachevitch, N. V., Blinov, P. D., Kotelnikov, E. V., Rubtsova, Y. V., Ivanov, V. V., Tutubalina, E.* (2015), Sentirueval: Testing Object-Oriented Sentiment Analysis Systems in Russian, Proceedings of the International Conference “Dialog 2015”, Moscow.
7. *Mikolov T., Chen K., Corrado G., Dean J.* (2013), Distributed Representations of Words and Phrases and their Compositionality, Proceedings of Advances in Neural Information Processing Systems 26 (NIPS 2013), Tahoe, pp. 3111–3119
8. *Mingbo M., Liang H., Bing X., Bowen Z.* (2015), Dependency-based Convolutional Neural Networks for Sentence Embedding, Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics (ACL-2015), pp. 174–179.
9. *Nivre J., Iomdin L. L., Boguslavsky I. M.* (2008), Parsing the SynTagRus Treebank of Russian, Proceedings of the 22nd International Conference on Computational Linguistics (Colling-2008), Manchester, pp. 641–648.
10. *Nivre, J., Hall, J., Nilsson J., Chanev A., Eryigit G., Kübler S., Marinov S.* (2005), MaltParser: A language-independent system for data-driven dependency parsing, Natural Language Engineering, Vol. 13, № 1 pp. 95–135
11. *Polyakov P. Y., Kalinina M. V., Pleshko V. V.* (2015), Automatic Object-Oriented Sentiment Analysis by Means of Semantic Templates and Sentiment Lexicon Dictionaries, Proceedings of the International Conference “Dialog 2015”, Moscow.
12. *Sachidanandan S., Sambaturu P., Karlapalem K.* (2013), NERTUW: Named entity recognition on tweets using Wikipedia, Concept Extraction Challenge Proceedings, Rio de Janeiro, pp. 67–70.