

Extracting Sentiment Attitudes from Analytical Texts via Piecewise Convolutional Neural Network

© N.L. Rusnachenko ¹

© N.V. Loukachevitch ²

¹ Bauman Moscow State Technical University, Moscow, Russia

² Lomonosov Moscow State University, Moscow, Russia

kolyarus@yandex.ru

louk_nat@mail.ru

Abstract. For deep text understanding, it is necessary to explore the connections between text units mentioning events, entities, etc. Depending on the further goals, it allows to consider the text as a graph of task-specific relations. In this paper, we focused on analysis of sentiment attitudes, where the attitude represents a sentiment relation from subject towards object. Given a mass media article and list of mentioned named entities, the task is to extract sentiment attitudes between them. We propose a specific model based on convolutional neural networks (CNN), independent of handcrafted NLP features. For model evaluation, we use RuSentRel 1.0 corpora, consisted of mass media articles written in Russian.

Keywords: sentiment analysis, convolutional neural networks, data intensive domains

1 Introduction

Automatic sentiment analysis, i.e. the identification of the authors' opinion on the subject discussed in the text, is one of the most popular applications of natural language processing during the last years.

One of the most popular direction becomes a sentiment analysis of user posts. Twitter [7] social network allows rapidly spread news in a form of short text messages, where some of them express user opinions. Such texts are limited in length and has only a single object for analysis – author opinion towards the service or product quality [1, 12]. These factors make this area well studied.

Large texts, such as analytical articles represent a complicated genre of documents for sentiment analysis. Unlike short posts, large articles expose a lot of entities where some of them connected by relations. The connectivity allows us to represent article as a graph. This kind of representation is necessary for information extraction (IE) [6]. Analytical texts contain Subject-Object relations, or *attitudes* conveyed by different subjects, including the author(s) attitudes, positions of cited sources, and relations of the mentioned entities between each other.

Besides, an analytical text can have a complicated discourse structure. Given an example: «*Donald Trump* _{e_1} accused *China* _{e_2} and *Russia* _{e_3} of “playing devaluation of currencies”». This sentence illustrates an attitude from subject e_1 towards multiple objects e_2 and e_3 , where objects have no attitudes within themselves. Additionally, statements of opinion can take several sentences, or refer to the entity mentioned several sentences earlier.

In this paper we introduce a problem of sentiment attitude extraction from analytical articles written in Russian. Here *attitude* denotes a directed relation from

subject towards an object, where each end of such relations represents a mentioned named entity.

We propose a model based on the modified architecture of Convolutional Network Networks (CNN). The model predicts a sentiment score for a given attitude in context. In case of the original CNN architecture, max pooling operation reduces information (convolved attitude context) quite rapidly. The modified architecture decreases the speed by reducing attitude context in pieces. The borders of such pieces related to attitude entities positions. We use RuSentRel 1.0 corpus for model evaluation. Both models based on original and modified CNN architectures significantly outperform baselines and perform better than classifiers based on handcrafted NLP features.

2 Related works

Relation extraction becomes popular since the appearance of the relation classification track in proceedings of SemEval-2010 conference. In [6] authors introduce a dataset for a task of *semantic classification between pair of common nominals*. The classification considered in terms of nominals context. This restriction introduced for simplicity and meaning disambiguation. The resulted model allows composing a semantic network for a given text with connections, accompanied by the relation type (Part-Whole, Member-Collection, etc.).

In 2014, the TAC evaluation conference in Knowledge Base Population (KBP) track included so-called sentiment track [5]. The task was to find all the cases where a query entity (sentiment holder) holds a sentiment (positive or negative) about another entity (sentiment target). Thus, this task was formulated as a query-based retrieval of entity-sentiment from relevant documents and focused only on query entities.

In [9] authors discover a *target sentiment* detection towards named entities in text. Depending on context, this sentiment arises from a variety of factors, such as

writer experience, attitudes from other entities towards target, etc.: «*So happy that [Kentucky lost to Tennessee]_{event}*». In latter example, Kentucky has negative attitude towards Tennessee, but the writer has positive one. The authors investigated how to detect named entity (NE) and sentiment expressed towards it. A variety of models based on conditional random fields (CRF) were implemented. All models were trained based on the list of predefined features. The experiments were subdivided into three tasks (in order of complexity growth): *NE recognition*, *subjectivity prediction* (fact of sentiment existence along the target), *sentiment NE prediction* (3-scale classification).

In [17] authors proceed discover of target sentiment detection. Being modeled as a sequence labeling problem, the authors exploit word embeddings with automatic features training within neural network models. Due to CRF model's affection, the authors experimented with models based on conditional neural fields architecture (CNF) [11]. As in [9], the task was considered in following parts: *entities classification*, *entities extraction and classification*.

MPQA 3.0 [4] is a corpus of analytical articles with annotated opinion expressions (towards entities and events). The annotation is sentence-based. For example, in the sentence «*When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet ...*», Imam is negative to Salman Rushdie, but is positive to the Prophet. The current corpus consists of 70 documents. In total, sentiments towards 4,459 targets are labeled.

The paper [3] studied the approach to the discovery of the documents attitudes between subjects mentioned in the text. The approach considers such features as relatedness between entities, frequency of a named entity in the text, direct-indirect speech, and other features. The best quality of opinion extraction obtained in the work was only about 36% F-measure by two sentiment classes, which illustrates the necessity of improving extraction of attitudes at the document level is significant.

For the analysis of sentiments with multiple targets in a coherent text, in the works [2] and [13] the concept of sentiment relevance is discussed. In [2], the authors consider several types of thematic importance of the entities discussed in the text: the main entity, an entity from a list of similar entities, accidental entity, etc. These types are treated differently in sentiment analysis of coherent texts.

For relation extraction, in [15] the task was modeled by convolutional neural network towards context representation based on word embedding features. Convoluting such embedding by a set of different filters, the authors implemented and trained Convolutional Neural Network (CNN) model for the relation classification task. Being applied for SemEval-2010 Task 8 dataset [6] the resulted model significantly outperforms the results of other participants.

However, for the relation classification task, the original max pooling reduces information extremely rapid, and hence, blurs significant relation aspects. The

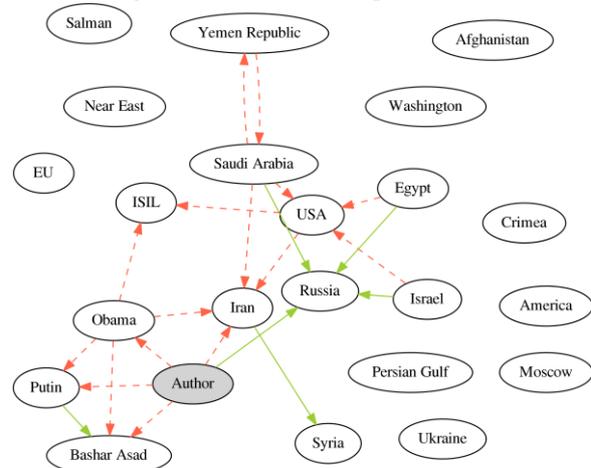
idea was proceeded by the authors of paper [16] in terms of *max pooling* operation. This operation applies for a convolved by filters data and extracts maximal value within each convolution. The authors proposed to treat each convolution in parts. The division into parts was related to attitude ends and was as follows: *inner*, and *outer*. This results in an advanced CNN architecture model and was dubbed as *Piecewise Convolutional Neural Network* (PCNN).

In this paper, we present an application of the PCNN model [16] towards sentiment attitudes extraction. We use automatically trainable features instead of handcrafted NLP features. For illustrating effectiveness, we compared our results with original CNN implementation, and other approaches: baselines, classifiers based on handcrafted features.

3 Dataset

We use RuSentRel 1.0 corpus¹ consisted of analytical articles from Internet-portal inosmi.ru [8]. These articles in the domain of international politics were obtained from foreign authoritative sources and translated into Russian. The collected articles contain both the author's opinion on the subject matter of the article and a large number of references mentioned between the participants of the described situations.

Figure 1 Opinion annotation example for article 4 (dashed: *negative* attitudes; solid: *positive* attitudes)



For the documents, the manual annotation of the sentiment attitudes towards the mentioned named entities has been carried out. The annotation can be subdivided into two subtypes:

- The author's relation to mentioned named entities;
- The relation of subjects expressed as named entities to other named entities.

Figure 1 illustrates annotated article attitudes in graph format. These opinions are as Subject-Object relations type in terms of related terminology [6] and recorded as triplets: (*Subject of opinion*, *Object of opinion*, *attitude*). The attitude can be negative (*neg*) or positive (*pos*), for example (*Author*, *USA*, *neg*), (*USA*, *Russia*, *neg*). Neutral opinions are not recorded. The attitudes are described for

¹ <https://github.com/nicolay-r/RuSentRel/tree/v1.0>

the whole documents, not for each sentence. In some texts, there were several opinions of the different sentiment orientation of the same subject in relation to the same object. This, in particular, could be due to the comparison of the sentiment orientation of previous relations and current relations (for example, between Russia and Turkey). Or the author of the article could mention his former attitude to some subject and indicate the change of this attitude at the current time. In such cases, it was assumed that the annotator should specify exactly the current state of the relationship. In total, 73 large analytical texts were labeled with about 2000 relations.

To prepare documents for automatic analysis, the texts were processed by the automatic name entity recognizer, based on CRF method [10]. The program identified named entities that were categorized into four classes: Persons, Organizations, Places and Geopolitical Entities (states and capitals as states). In total, 15.5 thousand named entity mentions were found in the documents of the collection. An analytical document can refer to an entity with several variants of naming (*Vladimir Putin – Putin*), synonyms (*Russia – Russian Federation*), or lemma variants generated from different wordforms. Besides, annotators could use only one of possible entity’s names describing attitudes. For correct inference of attitudes between named entities in the whole document, the dataset provides the list of variant names for the same entity found in our corpus. The current list contains 83 sets of name variants. This allows separating the sentiment analysis task from the task of named entity coreference.

A preliminary version of the RuSentRel corpus was granted to the Summer school on Natural Language Processing and Data Analysis², organized in Moscow in 2017. The collection was divided into the training and test parts. In the current experiments, we use the same division of the data. Table 1 contains statistics of the training and test parts of the RuSentRel corpus.

Table 1 Statistics of RuSentRel 1.0 corpus

Parameter	Training collection	Test collection
Number of documents	44	29
Sentences (avg. per doc.)	74.5	137
Mentioned NE (avg. per doc.)	194	300
Unique NE (avg. per doc.)	33.3	59.9
Pos. pairs of NE (avg. per doc.)	6.23	14.7
Neg. pairs of NE (avg. per doc.)	9.33	15.6
Neu. pairs of NE (avg. per doc.)	120	276
Avg. dist. between NE within a sentence in words	10.2	10.2
Share of attitudes expressed in a single sentence	76.5%	73%

The last line of the Table 1 shows the average number of named entities pairs mentioned in the same sentences

² <https://miem.hse.ru/clschool/>

without indication of any sentiment to each other per a document. This number is much larger than number of positive or negative sentiments in documents, which additionally stresses the complexity of the task.

4. Sentiment attitudes extraction

In this paper, the task of *sentiment attitude extraction* is treated as follows: given an attitude as a pair of its named entities, we predict a sentiment label of a pair, which could be *positive*, *negative*, or *neutral*.

The act of *extraction* is to select only those pairs, which were predicted as non-neutral. This leads to the following questions:

1. How to complete a set of all attitudes?
2. How to predict attitude labels?

4.1 Composing attitude sets

Given a list of synonym groups S provided by RuSentRel dataset (see Section 3), let $S(w)$ is a function which returns a synonym by given word³ or phrase w .

The pair of attitudes $a_1 = (e_{1,l}, e_{1,r})$ and $a_2 = (e_{2,l}, e_{2,r})$ are *equal up to synonyms* $a_1 \simeq a_2$ when both ends related to the same synonym group:

$$S(e_{1,l}) = S(e_{2,l}) \text{ and } S(e_{1,r}) = S(e_{2,r}) \quad (1)$$

Using Formula 1 we define that A is a set *without synonyms* as follows:

$$A: \nexists a_i, a_j \in A: \{a_i \simeq a_j, i \neq j\} \quad (2)$$

To complete a training set A_{train} , we first compose auxiliary sets without synonyms: A_s is a set of sentiment attitudes, and A_n – is a set of neutral attitudes. For A_s , the etalon opinions were used to find related named entities to compose sentiment attitudes. A_n consist of attitudes composed between all available named entities of the train collection. In this paper, the context attitudes were limited by a single sentence. Finally, completed A_{train} is an expansion A_s with A_n :

$$A_{train} = A_s \cup A_n: \nexists i, j: \{a_i \simeq a_j, a_i \in A_s, a_j \in A_n\} \quad (3)$$

To estimate the model, we complete the test set A_{test} of neutral attitudes without synonyms. It consists of attitudes composed between all available named entities within a single sentence of the test collection. Table 2 illustrates amount of attitudes both for the train and test collections.

Table 2 Context attitudes amount

Attitudes count	A_{train}	A_{test}
Positive	571 (7.2%)	-
Negative	735 (9.3%)	-
Neutral	6584 (83.5%)	8024

4.2 Labels prediction

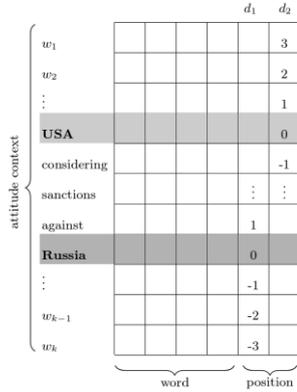
For label prediction, we use an approach that exploits a word embedding model and automatically trainable features. We implemented an advanced CNN model, dubbed as Piecewise Convolutional Neural Network

³ The case of synonym absence has been resolved by completing a new group with the single element $\{w\}$.

(PCNN), proposed by [16].

4.2.1 Attitude embedding

The attitude embedding is a form of an attitude representation in a way of a related context, where each **Figure 1** Attitude embedding matrix



word of a context is an embedding vector. Figure 1 illustrates a context for an attitude with “USA” and “Russia” as named entities: «...USA is considering the possibility of new sanctions against Russia...».

Picking a context that includes attitude entities with the inner part, we expand it with words by both sides equally and finally composing a text sample $s = \{w_1, \dots, w_k\}$ of a size k . Additionally, each w_i has been lowercased and lemmatized.

Let E_w is a precomputed embedding vocabulary, which we use to compose word embeddings \mathbf{e}_{w_i} . Each w_i might be a part of an *attitude entity* or a *text*. In the latter case $\mathbf{e}_{w_i} = E_w(w_i)$ ⁴. For attitude entities, we consider them as single words. Due to that some entities are phrases (for example “Russian Federation”), the embedding for them calculated as a sum of each component word w_j in the phrase:

$$\mathbf{e}_{w_i} = E_w(w_j) \quad (4)$$

Given a sample s , for each word w_i of it, we compose vector \mathbf{w}_i as a concatenation of vectors \mathbf{e}_{w_i} (word) and a pair of distances (d_1, d_2) (position) related to each entity⁵. Given a one attitude entity e_1 , we let $d_{1,i} = pos(w_i) - pos(e_1)$, where $pos(\cdot)$ is a position index in sample s by a given argument. The same computations are applied for $d_{2,i}$ with the other entity e_2 respectively. Composed $E_a = \{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ represents an attitude embedding matrix.

4.2.2 Convolution

This step of data transformation applies filters towards the attitude embedding matrix (see Figure 2). Treating the latter as a feature-based attitude representation, this approach implements feature merging by sliding a filter of a fixed size within a data and transforming information in it.

According to Section 4.2.1, $E_a \in \mathbb{R}^{k \times m}$ is an attitude embedding matrix with a text segment of size k and vector size m . We regard E_a as a sequence of rows

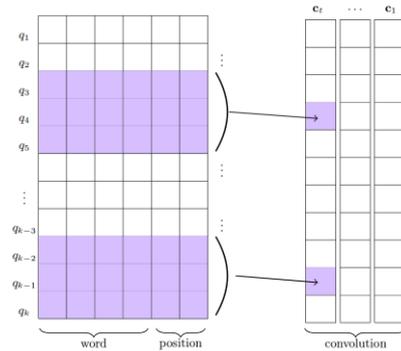
⁴ In case of word absence w_i in E_w , the zero vector was used

$Q = \{q_1, \dots, q_k\}$, where $q_i \in \mathbb{R}^m$. We denote $q_{i:j}$ as consequent vectors concatenation from i 'th till j 'th positions. An application of $\mathbf{w} \in \mathbb{R}^d$, ($d = w \cdot m$) towards the concatenation $q_{i:j}$ is a sequence *convolution* by filter \mathbf{w} , where w is a filter window size. Figure 1 illustrates $w = 3$. For convolving calculation c_j , we apply scalar multiplication as follows:

$$c_j = \mathbf{w}q_{j-w+1:j} \quad (5)$$

Where $j \in \overline{1 \dots k}$ is filter offset within the sequence Q . We decide to let q_i a zero-based vector of size m in case when $i < 0$ or $i > k$. As a result, $\mathbf{c} = \{c_1, \dots, c_k\}$ with shape $\mathbf{c} \in \mathbb{R}^k$ is a convolution of a sequence Q by filter w .

Figure 2 Convolving embedding matrix example



To get multiple feature combinations, a set of different filters $W = \{\mathbf{w}_1, \dots, \mathbf{w}_t\}$ has been applied towards the sequence Q , where t is an amount of filters. This leads to a modified Formula 1 by introduced layer index i as follows:

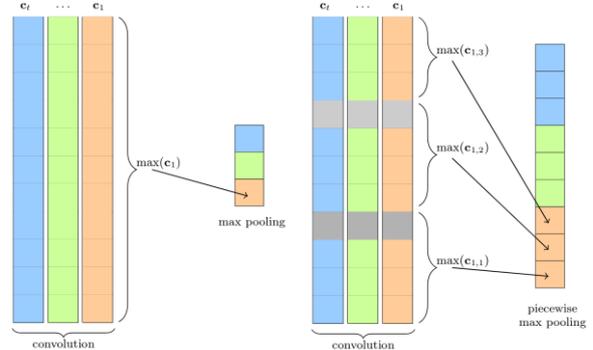
$$c_{i,j} = \mathbf{w}_i q_{j-w+1:j} \quad (6)$$

Denoting $\mathbf{c}_i = \{c_{i,1}, \dots, c_{i,n}\}$ in Formula 1 we reduce the latter by index j and compose a matrix $C = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_t\}$ which represents convolution matrix with shape $C \in \mathbb{R}^{k \times t}$. Figure 1 illustrates an example of convolution matrix with $t = 3$.

4.2.3 Max pooling

Max pooling is an operation that reduces values by keeping maximum. In original CNN architecture, max pooling applies separately per each convolution $\{\mathbf{c}_1, \dots, \mathbf{c}_t\}$ of t layers (see Figure 3, left).

Figure 3 Max pooling comparison (left: original CNN max pooling; right: piecewise version)



It reduces convolved information quite rapidly, and therefore is not appropriate for attitude classification task. To keep context aspects that are inside and outside of the attitude entities, authors [16] perform *piecewise max pooling*. Given attitude entities as borders, we divide each c_i into inner, left and right segments $\{c_{i,1}, c_{i,2}, c_{i,3}\}$ (see Figure 3, *right*). Then max pooling applies per each segment separately:

$$p_{i,j} = \max(c_{i,j}), i \in \overline{1 \dots t} \quad j \in \overline{1 \dots 3} \quad (6)$$

Thus, for each c_i we have a $\mathbf{p}_i = \{p_{i,1}, p_{i,2}, p_{i,3}\}$. Concatenation of these sets $\mathbf{p}_{i,j}$ results in $\mathbf{p} \in \mathbb{R}^{3t}$ and that is a result of piecewise max pooling operation. At the last step we apply the hyperbolic tangent activation function. The shape of resulted \mathbf{d} remains unchanged:

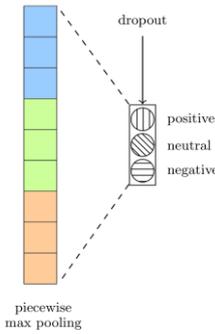
$$\mathbf{d} = \tanh(\mathbf{p}), \quad \mathbf{d} \in \mathbb{R}^{3t} \quad (7)$$

4.2.4 Sentiment Prediction

Before we receive a neural network output, the result $\mathbf{d} \in \mathbb{R}^{3t}$ of the previous step passed through the fully connected hidden layer:

$$o = W_1 \mathbf{d} + b, W_1 \in \mathbb{R}^{c \times 3t}, b \in \mathbb{R}^c \quad (8)$$

Figure 4 Max pooling transformation



In Formula 8, c is an expected amount of classes, and o is an output vector. The elements of the latter vectors are unscaled values. We use a *softmax* transformation to obtain probabilities per each output class. Figure 4 illustrates a 3-dimensional output vector. To prevent a model from overfitting, we employ *dropout* for output neurons during training process.

4.2.5 Training

As a function, the implemented neural network model depends on the parameters divided into the following groups: I represents an input for supervised learning, and H describes hidden states that are trainable during network optimization. Formula 9 illustrates network θ function dependencies:

$$\theta = (I; H) = (T; W, W_1, b) \quad (9)$$

The group of input parameters I consist of m tuples $T = \{t_1, \dots, t_m\}$, where $t_i = (A_e, y)$ includes attitude embedding A_e with the related label $y \in \mathbb{R}^c$. The group of hidden parameters H includes a set of convolution filters W , hidden fully connected layer W_1 and bias

vector b .

The neural network training process includes the following steps:

1. Split T into list of batches $B = \{t_1, \dots, t_q\}$ with the fixed size of q , where $t_i \in T$;
2. Randomly choose b_s from list of batches B to perform a forward propagation through the network and receive $o_s = \{o_1, \dots, o_q\} \in \mathbb{R}^{q \cdot c}$;
3. Given an o_s we compute *cross entropy loss* as follows:

$$J(\theta) = \sum_{j=1}^c \log p(y_i | o_{i,j}; \theta), i \in \overline{1 \dots q} \quad (10)$$

4. Update hidden variables H of θ using the calculated gradients from the previous step;
5. Repeat steps 2-4 while the necessary epoch count will not be reached.

5 Experiments

We consider attitudes as a pair of named entities within a single sentence (see Section 4.1). The distance in words within pair was limited by segment size $k = 50$. According to Table 1 (see “Share of attitudes expressed in a single sentence”) it allows us to cover up to 76.5% and 74% of sentiment attitudes for the train and test collections respectively. Table 2 illustrates an amount of extracted attitudes from train and test collections.

To select an embedding model E_w , the average distance between attitude entities was taken into account. According to Table 1 (see «avg. dist. between NE within a sentence in words»), we were interested in a Skip-gram based model which covers our estimation. We use a precomputed and publicly available word2vec⁶ model⁷ based on news articles with window size of 20 and vector size of 1000. To perform text lemmatization, we utilize Yandex Mystem⁸.

We use the *adadelta* optimizer for model training with parameters that were chosen according to [14]. For dropout probability, the statistically optimal value for most classification tasks was chosen.

For model evaluation, we use $F_1(P, N)$ -macro measure. It combines *recall* and *precision* both by positive (P) and negative (N) classes. We experimentally study the effectiveness of a model by varying *filters count*.

Table 3 illustrates the results for both implemented PCNN model⁹ and the original CNN model in runs, where each run varies in terms of settings. Due to that $J(\theta)$ has a non-convex shape with large amount of local minimums, and initial hidden state varies by each we provide multiple evaluation results during the training process at certain epochs $F_1(e)$, where e is an amount of epochs were passed. According to the obtained results (see Table 3), we may conclude that using greater amount of filters allows to accelerate training process for

⁶ <https://code.google.com/p/word2vec/>

⁷ http://rusvectors.org/static/models/rusvectors2/news_myst_em_skipgram_1000_20_2015.bin.gz

⁸ <https://tech.yandex.ru/mystem/>

⁹ github.com/nicolay-r/sentiment-pcnn/tree/damdid-2018

Table 3 $F_1(P,N)$ results through epochs for CNN and PCNN models

<i>model</i>	<i>filters count</i>	$F_1(25)$	$F_1(50)$	$F_1(75)$	$F_1(100)$	$F_1(150)$	$F_1(200)$	$F_1(250)$
CNN	100	0.03	0.07	0.08	0.13	0.20	0.25	0.29
	200	0.06	0.11	0.15	0.19	0.17	0.25	0.26
	300	0.12	0.18	0.24	0.28	0.31	0.30	0.30
PCNN	100	0.06	0.13	0.23	0.21	0.28	0.29	0.30
	200	0.17	0.24	0.29	0.29	0.29	0.30	0.31
	300	0.19	0.27	0.29	0.29	0.29	0.29	0.29

both models. Comparing original CNN with the Piecewise version, the model of the latter architecture reaches top results ($F_1(P,N) \geq 0.30$) significantly faster. According to Table 4, proposed approach significantly outperforms the baselines and performs better than conventional classifiers [8]. Manually implemented feature set was used to train KNN, SVM, Naive Bayes, and Random Forest classifiers [8]. For the same dataset, SVM and Naive Bayes achieved 16% by F-measure, and the best result has been obtained by the Random Forest classifier (27% F-measure). To assess the upper bound for experimented methods, the expert agreement with etalon labeling was estimated (Table 4, last row). Overall, we may conclude that this task still remains complicated and the results are quite low. It should be noted that the authors of the [3], who worked with much smaller documents written in English, reported F-measure 36%.

Table 4 Experiment results

<i>Method</i>	<i>Precision</i>	<i>Recall</i>	$F_1(P,N)$
Neg	0.03	0.39	0.05
Pos	0.02	0.40	0.04
Distr	0.05	0.23	0.08
School	0.13	0.10	0.12
KNN	0.18	0.06	0.09
SVM (Grid)	0.09	0.36	0.15
Random forest	0.41	0.21	0.27
CNN	0.41	0.23	0.31
PCNN	0.42	0.23	0.31
Expert agreement	0.62	0.49	0.55

5 Conclusion

This paper introduces the problem of sentiment attitude extraction from analytical articles. The key point of the proposed solution that it does not depend on handcrafted feature implementation. The models based on the Convolutional Neural Network architecture were used.

In the current experiments, the problem of sentiment attitude extraction is considered as a three-class machine learning task. We experimented with CNN-based models by studying their effectiveness depending on convolutional filters count. Increasing the latter parameter accelerates training process. Comparing original architecture with the piecewise modification, the model of the latter reaches better results faster. Both models significantly outperform the baselines and

perform better than approaches based on handcrafted features.

Due to the dataset limitation and manual annotating complexity, in further works we plan to discover unsupervised pre-training techniques based on automatically annotated articles of external sources. In addition, the current attitude embedding format has no information about related article in whole, which is an another direction of further improvements.

References

- [1] Alimova, I., Tutubalina, E.: Automated detection of adverse drug reactions from social media posts with machine learning. In: Proceedings of International Conference on Analysis of Images, Social Networks and Texts, pp. 1-12, (2017)
- [2] Ben-Ami, Z., Feldman, R., Rosenfeld, B.: Entities' sentiment relevance. ACL-2013, 2, pp. 87–92, (2014)
- [3] Choi, E., Rashkin, H., Zettlemoyer, L., Choi, Y.: Document-level sentiment inference with social, faction, and discourse context. In: Proceedings of the 54th annual meeting of the association for computational linguistics. ACL, pp. 333–343, (2016)
- [4] Deng, L., Wiebe, J.: MPQA 3.0: An entity/event-level sentiment corpus. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1323–1328, (2015)
- [5] Ellis, J., Getman, J., Strassel, S., M.: Overview of linguistic resources for the TAC KBP 2014 evaluations: Planning, execution, and results. Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology, pp. 17–18, (2014)
- [6] Hendrickx, I., et. al.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, Association for Computational Linguistics, pp. 94–99, (2009)
- [7] Loukachevitch, N., Rubtsova Y.: Sentirueval-2016: Overcoming time gap and data sparsity in tweet sentiment analysis. In: Computational Linguistics and Intellectual Technologies Proceedings of the Annual International

- Conference Dialogue, Moscow, RGGU, pp. 416-427, (2016)
- [8] Loukachevitch, N., Rusnachenko, N.: Extracting sentiment attitudes from analytical texts. In: Proceedings of International Conference of Computational Linguistics and Intellectual Technologies Dialog-2018, pp. 455–464, (2018)
- [9] Mitchell, M., Aguilar, J., Wilson, T., Van Durme, B.: Open domain targeted sentiment. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1643–1654, (2013)
- [10] Mozharova, V., Loukachevitch, N.: Combining knowledge and CRF-based approach to named entity recognition in Russian. In: International Conference on Analysis of Images, Social Networks and Texts, pp. 185–195, (2016)
- [11] Peng, J., Bo, L., Xu, J.: Conditional neural fields. In: Advances in neural information processing systems, pp. 1419–1427, (2009)
- [12] Rosenthal, S., Farra, N., Nakov, P.: Semeval-2017 task 4: Sentiment analysis in twitter. In: Proceedings of SemEval-2017 workshop, pp. 502-518, (2017)
- [13] Scheible, C., Schutze, H.: Sentiment relevance. In: Proceedings of ACL 2013 1, pp. 954–963, (2013)
- [14] Zeiler, M.D.: Adadelata: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
- [15] Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344, (2014)
- [16] Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1753–1762, (2015)
- [17] Zhang M., Zhang Y., Vo D. T.: Neural networks for open domain targeted sentiment. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 612-621, (2015)