EPiC Series in Language and Linguistics

Volume 4, 2019, Pages 1–10

Proceedings of Third Workshop "Computational linguistics and language science"

EPiC
Language
and Linguistics

Using Convolutional Neural Networks for Sentiment Attitude Extraction from Analytical Texts

N.L. Rusnachenko[1] and N.V. Loukachevitch[2]

[1] Bauman Moscow State Technical University, Moscow, Russia
kolyarus@yandex.ru

[2] Lomonosov Moscow State University, Moscow, Russia
louk_nat@mail.ru

Abstract

In this paper we present an application of the specific neural network model for sentiment attitude extraction without implementing handcrafted features. Given a mass-media article with the list of named entities mentioned in it, the task is to extract sentiment relations between these entities. We consider this problem as a three-class machine learning task for the whole documents. The modified architecture of the Convolutional Neural Networks was used and called as Piecewise Convolutional Neural Network (PCNN). The latter exploits positions of named entities in texts to emphasize aspects for inner and outer contexts of extracted relation. For the experiments, the RuSentRel corpus was used, it contains Russian analytical texts in the domain of international relations.

1 Introduction

Automatic sentiment analysis, i.e. the identification of the author's 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 studied text genres in the sentiment analysis task is users' reviews about products or services. Another popular type of texts for sentiment analysis are short messages posted in social networks, such as Twitter [7]. Such texts are limited in length and usually discuss a single entity (but, perhaps in its various aspects), and the opinion is expressed by author of review [1, 10].

One of the most complicated genres of documents for sentiment analysis are analytical articles that analyze a situation in some domain, for example, politics or economy. These texts contain opinions conveyed by different subjects, including the author(s)' attitudes, positions of cited sources, and relations of the mentioned entities between each other.

Analytical texts usually mention a lot of named entities, and only a few of them are subjects or objects of sentiment attitudes. Besides, an analytical text can have a complicated discourse structure. Statements of opinion can take several sentences, or refer to the entity mentioned several sentences earlier. Also a sentence containing an opinion about a specific target may also include other named entities, which complicates the recognition of sentiment attitudes, their subjects and objects.

In this paper we describe a problem of sentiment attitude extraction related to large mass-media analytical articles written in Russian. Considering «attitudes» as directed relations between *named entities* (NE), the task is to extract only sentiment of them. Traditionally, such an extraction process relies on a set of features of inner/outer context for a given attitude. The latter leads to a handcrafting process, which sometimes may be complicated because of a large amount of necessary features.

The model proposed in this paper does not depends on the manually created features. This has been achieved by using the Convolutional Neural Networks (CNN) architecture which exposes filters as automatically trainable features. We experiment with an advanced CNN architecture which transforms convolved information by pieces. For model evaluation and training the RuSentRel corpora was used. This task can be considered as a specific subtask of relation extraction.

2 Related Work

The task of attitude recognition toward named entities or events including opinion holder identification from full texts did not attract much attention. 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 positive or negative sentiment 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[1].

In Deng and Wiebe (2015) [4], MPQA 3.0 corpus is described. In the corpus, sentiments towards entities and events are labeled. 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.

Choi et al. (2016) [3] studied the approach to the recovery 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, which shows that the necessity of improving extraction of attitudes at the document level is significant and this problem has not been sufficiently studied.

For the analysis of sentiments with multiple targets in a coherent text, in the works [2, 11] the concept of sentiment relevance is discussed. Ben-Ami et al. (2014) [2] 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.

Each attitude may be considered in terms of related article context. The context consist of words and may be treated as an embedding matrix with word vectors as rows. Convolving such embeddings by a set of different filters, Zeng et al. (2014) [14] implemented and trained the Convolutional Neural Network (CNN) model for the relation classification task. Being applied for the SemEval-2010 Task 8 dataset [6], the obtained model significantly outperformed the results of other participants.

This idea was further proceeded by Zeng et al. (2015) in terms of *max pooling* operation [13]. This is an operation, which is applied to the convolved by filters data and extracts the maximal values within each convolution. However, for the relation classification task, original

[1] https://tac.nist.gov/2014/KBP/Sentiment/index.html

max pooling reduces information extremely rapid, and hence, blurs significant relation aspects. Authors proposed to treat each convolution in parts. The division into parts was related to attitude entities and was as follows: *inner* (between entities), and *outer*. The latter represents two separated pieces: before the first entity appeared in the sentence, and after the last entity appeared in the sentence. This approach results in an advanced CNN architecture model and was dubbed as «Piecewise Convolutional Neural Network» (PCNN).

In this paper, we describe a model that based on automatically trained and formed features. The CNN architecture was chosen as a framework, where the input convolved by different *filters*, where each filter acts as an unique feature. To emphasize attitude position in text and reduce the convolution speed, we implemented and experiment with PCNN model [13].

3 Corpus and Annotation

We use RuSentRel v1.0 corpus[2] consisted of analytical articles from Internet-portal inosmi.ru [8]. These articles in the domain of international politics are 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 attitudes mentioned between the participants of the described situations.

For the documents, the manual annotation of the sentiment attitudes towards the mentioned named entities have been carried out. The annotation divided into two subtypes:

1. The author's relation to mentioned named entities;

2. The relation of subjects expressed as named entities to other named entities.

These opinions were recorded as triples: (*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 or lack of opinions are not recorded. Attitudes are described for 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 [9]. 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, corpora 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.

[2] https://github.com/nicolay-r/RuSentRel/tree/v1.0

Table 1: Statistics of RuSentRel v1.0 corpus

Parameter	Training collection	Test collection
Number of documents	44	29
Avg. number of sentences per doc.	74.5	137
Avg. number of mentioned NE per doc.	194	300
Avg. number of unique NE per doc.	33.3	59.9
Avg. number of positive pairs of NE per doc.	6.23	14.7
Avg. number of negative pairs of NE per doc.	9.33	15.6
Avg. dist. between NE within a sentence in words	10.2	10.2
Share of attitudes expressed in a single sentence	76.5%	73%
Avg. number of neutral pairs of NE per doc.	120	276

A preliminary version of the RuSentRel v1.0 corpus was granted to the Summer school on Natural Language Processing and Data Analysis[3], 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 related parts of the RuSentRel corpus. The last line of the Table shows the average number of named entities pairs mentioned in the same sentences 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 Piecewise Convolutional Neural Network

For label prediction, we use an approach that does not depend on handcrafted features. We implemented an advanced CNN model proposed by [13] with an architecture presented in Figure 1. Besides architecture details and transformation aspects, it illustrates a dataflow for an attitude with «USA» and «Russia» as a named entities: *The US is considering the possibility of new sanctions against Russia*[4]. Next, we describe each architectural block in details.

4.1 Attitude embedding

Attitude embedding is a form of an attitude representation in a way of a related context, where each word of a context is an embedding vector. 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 $s = \{w_1, ..., w_k\}$ 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 $\mathbf{e}_{w_i} = E_w(w_i)$[5]. 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:

$$\mathbf{e}_{w_i} = \sum_j E_w(w_j) \tag{1}$$

[3] https://miem.hse.ru/clschool/

[4] Figure 1 illustrates a simplified context of an attitude example for a clarity purposes

[5] In case of word absence w_i in E, the zero vector was used

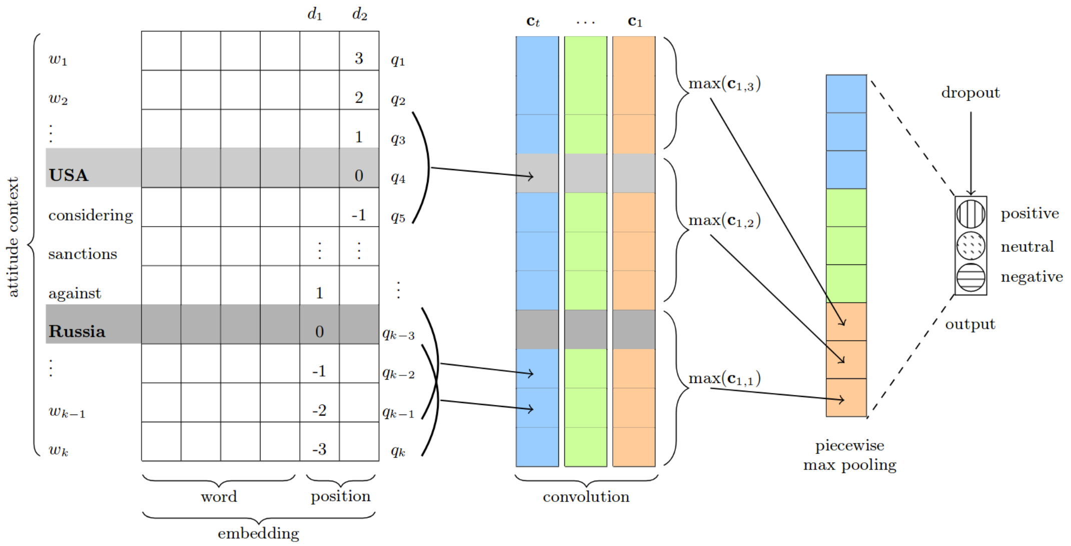

Figure 1: Piecewise Convolutional Neural Network

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 $\langle d_1, d_2 \rangle_i$ (position) related to each entity (see Figure 1). 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 $E_a = \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ represents an attitude embedding matrix.

4.2 Convolution

This step of data transformation applies filters towards the attitude embedding matrix. 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 4.1, $E_a \in \mathbb{R}^{k \times m}$ 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 $Q = \{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$, where $\mathbf{q}_i \in \mathbb{R}^m$. We denote $\mathbf{q}_{i:j}$ 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 $\mathbf{q}_{i:j}$ 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:

$$c_j = \mathbf{w}\mathbf{q}_{j-w+1:j} \tag{2}$$

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

To get multiple feature combinations, a set of different filters $W = \{\mathbf{w_1}, \ldots \mathbf{w_t}\}$ has been applied towards the sequence Q, where t is an amount of filters. This leads to a modified Formula 2 by introduced layer index i as follows:

$$c_{i,j} = \mathbf{w}_i\mathbf{q}_{j-w+1:j} \tag{3}$$

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

4.3 Piecewise Max Pooling

Max pooling is an operation that reduces values by keeping maximum. In original CNN architecture, max pooling applies separately per each convolution $\{\mathbf{c}_1, \ldots, \mathbf{c}_t\}$ of t layers. 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 [13] perform *piecewise max pooling*. Given attitude entities as borders, we divide each c_i into inner, left and right segments $\{\mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \mathbf{c}_{i,3}\}$. Then max pooling applies per each segment separately:

$$p_{i,j} = max(\mathbf{c}_{i,j}), \quad i \in \overline{1 \ldots t}, \quad j \in \{1, 2, 3\} \tag{4}$$

Thus, for each $\mathbf{c}_i$ we have a set $\mathbf{p}_i = \{p_{i,1}, p_{i,2}, p_{i,3}\}$. 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 d remains unchanged:

$$d = tanh(\mathbf{p}), \in \mathbb{R}^{3t} \tag{5}$$

4.4 Sentiment prediction

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

$$o = W_1 d + b, \qquad W_1 \in \mathbb{R}^{c \times 3t}, \quad b \in \mathbb{R}^c \tag{6}$$

Where 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 1 illustrates a 3-dimentional output vector. To prevent a model from overfitting, we employ *dropout* for output neurons during training process.

4.5 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 7 illustrates network θ function dependencies:

$$\theta = \langle I; H \rangle = \langle T; W, W_1, b \rangle \tag{7}$$

The group of input parameters I consist of m tuples $T = \{t_1, \ldots, t_m\}$, where $t_i = \langle A_e, y \rangle$ 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 $B = \{t_1, \ldots, t_q\}$ 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 $o_s = \{o_1, \ldots, o_q\} \in \mathbb{R}^{q \cdot c}$;

3. Given an o_s we compute *cross entropy loss* as follows:

$$J(\theta) = \sum_{j=1}^{c} \log p(y_i | o_{i,j}; \theta), \; i \in \overline{1 \ldots q} \tag{8}$$

4. Update hidden variables H of θ using the calculated gradients from the previous step;

5. Repeat steps 2-4 while the necessary epochs count will not be reached.

5 Experiments

In the current experiment we consider the problem of extracting sentiment relations from analytical texts as a three-class supervised machine learning task. As a measure of classification quality, we take the averaged Precision, Recall and F-measure of positive and negative classes. All the named entities mentioned in a document are grouped in pairs: (NE_1, NE_2), (NE_2, NE_1). All the generated pairs should be classified as having positive, negative, or neutral sentiment from the first named entity of the pair (opinion holder) to the second entity of the pair (opinion target). To support this task, we added neutral sentiments for all pairs not mentioned in the annotation and co-occurred in the same sentences into the training and test collections.

Table 2 illustrates the predefined settings. We use 44 documents as a training collection, and 29 documents as a test collection (see Table 1) in the same manner as the data were provided for the Summer School mentioned in Section 3. We generate and classify only those pairs of named entities that co-occur in the same sentence at least once in a document, which distance in words within pair was limited by $k = 50$. According to Table 1 it allows us to cover up to 76.5% and 74% of sentiment attitudes for the train and test collections respectively.

To select an embedding model E_w, the average distance between attitude entities were 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 word2vec[6] model[7] based on news articles with the window size of 20 and vector size of 1000. We also utilize Yandex Mystem[8] for text lemmatization.

We use the *adadelta* optimizer for model training with parameters that were chosen according to Zeiler and Matthew (2012) [12]. For dropout probability, the statistically optimal value for most classification tasks was chosen.

Table 2: Predefined training parameters

batch size	*segment size*	*window size*	*embedding params*	*adadelta params*	*dropout probability*
50	$k = 50$	$w = 3$	$window = 20, \; \lvert v \rvert = 1000$	$\rho = 0.95, \; \epsilon = 10^{-6}$	$\rho = 0.5$

Table 3 illustrates the results of several baselines towards the test collection:

1. `baseline_neg` – all pairs of named entities are labeled as negative;

2. `baseline_pos` – all pairs are labeled as positive;

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

[7] http://rusvectores.org/static/models/rusvectores2/news_mystem_skipgram_1000_20_2015.bin.gz

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

Table 3: Baselines for sentiment extraction between named entities for RuSentRel corpus.

method	*precision*	*recall*	$F_1(P, N)$
baseline_neg	0.03	0.39	0.05
baseline_pos	0.02	0.40	0.04
baseline_random	0.04	0.22	0.07
baseline_distr	0.05	0.23	0.08
baseline_school	0.13	0.10	0.12

3. `baseline_random` – the pairs are labeled randomly;
4. `baseline_distr` – the pairs are labeled randomly according to the sentiment distribution in the training collection;
5. `baseline_school` – the results obtained by the best team at the Summer school[9].

We experimentally studied the effectiveness of a model by varying an amount of convolutional filters count $c \in [100, 150, 200, 250]$. Table 4 illustrates the results for the both implemented[10] original and piecewise CNN models in runs, where each run varies in terms of different filters count. For training process assessment, we perform multiple F-macro evaluations $F_1(e)$ during model training, where e is an amount of epochs were passed.

Table 4: $F_1(P, N)$ results through epochs for CNN and PCNN models

model	c	$F_1(25)$	$F_1(50)$	$F_1(75)$	$F_1(100)$	$F_1(150)$	$F_1(200)$	$F_1(250)$
CNN	100	0.03	0.07	0.08	0.13	0.20	0.25	0.29
	150	0.06	0.09	0.10	0.11	0.19	0.17	0.22
	200	0.06	0.11	0.15	0.19	0.17	0.25	0.26
	250	0.16	0.18	0.23	0.27	**0.30**	0.27	**0.30**
PCNN	100	0.06	0.13	0.23	0.21	0.28	0.29	**0.30**
	150	0.06	0.20	0.24	0.29	0.27	**0.30**	0.28
	200	0.17	0.24	0.29	0.29	0.29	**0.30**	**0.31**
	250	0.17	0.26	0.29	0.28	0.29	**0.30**	**0.30**

According to the obtained result we may conclude that increasing amount of convolution filters allows us to accelerate training process for both models. Comparing results of CNN with the piecewise version, the latter reaches the range $F_1(P, N) \geq 0.30$ faster. The proposed approach significantly outperforms the baselines and performs better than feature dependent conventional approaches [8]. For latter, manually implemented feature set was used to train KNN, SVM, Naive Bayes, and Random Forest classifiers. Using the same dataset, SVM and Naive Bayes achieved 16% F-measure, and the best result has been obtained by the Random Forest classifier (27% F-measure). Overall, we may conclude that this task still remains complicated and the results are quite low. It should be noted that Choi et al. (2016) [3], who worked with much smaller documents written in English, reported F-measure 36%.

[9] https://miem.hse.ru/clschool/results

[10] https://github.com/nicolay-r/sentiment-pcnn/tree/clls-2018

Conclusion

This paper introduces the problem of sentiment attitude extraction from mass-media articles. The keypoint 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 in case of a different amount of convolutional filters. It was shown that the use of modified architecture allows us to accelerate the training process. The proposed models significantly outperform the baselines and perform better than conventional classifiers.

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. Also current attitude embedding format has no information of entire article, which is an another direction of further improvements.

Acknowledgements

This work is partially supported by RFBR grant N 16-29-09606.

References

[1] I. Alimova and E. Tutubalina. 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*, pages 1–12, 2017.

[2] Z. Ben-Ami, R. Feldman, and B. Rosenfeld. Entities' sentiment relevance. *ACL-2013*, 2:87–92, 2014.

[3] Eunsol Choi, Hannah Rashkin, Luke Zettlemoyer, and Yejin Choi. Document-level sentiment inference with social, faction, and discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 333–343, 2016.

[4] L. Deng and J. Wiebe. 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*, pages 1323–1328, 2015.

[5] J. Ellis, J. Getman, and M. Strassel, S. 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*, pages 17–18, 2014.

[6] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 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*, pages 94–99. Association for Computational Linguistics, 2009.

[7] N. Loukachevitch and V. Rubtsova Y. Sentirueval-2016: Overcoming time gap and data sparsity in tweet sentiment analysis. *Computational Linguistics and Intellectual Technologies Proceedings of the Annual International Conference Dialogue, Moscow, RGGU*, pages 416–427, 2016.

[8] N. Loukachevitch and N. Rusnachenko. Extracting sentiment attitudes from analytical texts. *Proceedings of International Conference of Computational Linguistics and Intellectual Technologies Dialog-2018*, 2018.

[9] A. Mozharova, V. and V. Loukachevitch, N. Combining knowledge and crf-based approach to named entity recognition in russian. *International Conference on Analysis of Images, Social Networks and Texts*, pages 185–195, 2016.

[10] S. Rosenthal, N. Farra, and P Nakov. Semeval-2017 task 4: Sentiment analysis in twitter. *In Proceedings of SemEval-2017 workshop*, pages 502–518, 2017.

[11] C. Scheible and H. Schütze. Sentiment relevance. *In Proceedings of ACL 2013*, 1:954–963, 2013.

[12] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[13] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015.

[14] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, 2014.