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

Moscow, June 1–4, 2016

COMPARISON OF NEURAL NETWORK ARCHITECTURES FOR SENTIMENT ANALYSIS OF RUSSIAN TWEETS

Arkhipenko K. (arkhipenko@ispras.ru)[1,2],
Kozlov I. (kozlov-ilya@ispras.ru)[1,3],
Trofimovich J. (integral@ispras.ru)[1],
Skorniakov K. (kirill.skorniakov@ispras.ru)[1,3],
Gomzin A. (gomzin@ispras.ru)[1,2],
Turdakov D. (turdakov@ispras.ru)[1,2,4]

[1]Institute for System Programming of RAS, Moscow, Russia

[2]Lomonosov Moscow State University, CMC faculty, Moscow, Russia

[3]MIPT, Dolgoprudny, Russia

[4]FCS NRU HSE, Moscow, Russia

The paper presents evaluation of three neural network based approaches to Twitter sentiment analysis task performed at SentiRuEval-2016. The task focuses on sentiment classification of tweets about banks and telecommunication companies.

Our team submitted three solutions which are based on different supervised classifiers: Gated Recurrent Unit neural network (GRU), convolutional neural network (CNN), and SVM classifier with domain adaptation combined with previous two classifiers. We used vector representations of words obtained with word2vec model as features for classifiers. These classifiers were trained on labeled data provided by organizers of the evaluation. Additionally, we collected several million posts and comments from social networks for training word2vec model.

According to evaluation results, GRU-based solution shows the best macro-averaged F1-score for both domains (banks and telecommunication companies) and also has the best micro-averaged F1-score for banks domain among all solutions submitted to SentiRuEval.

Key words: sentiment analysis, opinion mining, recurrent neural network, convolutional neural network

СРАВНЕНИЕ АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ АНАЛИЗА ТОНАЛЬНОСТИ РУССКОЯЗЫЧНЫХ ТВИТОВ

Архипенко К. (arkhipenko@ispras.ru)[1,2],
Козлов И. (kozlov-ilya@ispras.ru)[1,3],
Трофимович Ю. (integral@ispras.ru)[1],
Скорняков К. (kirill.skorniakov@ispras.ru)[1,3],
Гомзин А. (gomzin@ispras.ru)[1,2],
Турдаков Д. (turdakov@ispras.ru)[1,2,4]

[1]Институт Системного Программирования РАН, Москва, Россия

[2]Московский государственный университет им. М. В. Ломоносова, ВМК, Москва, Россия

[3]Московский физико-технический институт, Долгопрудный, Россия

[4]НИУ Высшая школа экономики, ФКН, Москва, Россия

Ключевые слова: анализ тональности, извлечение мнений, рекуррентная нейронная сеть, свёрточная нейронная сеть

Introduction

The paper describes participation in SentiRuEval-2016 competition. The task of the competition focuses on object-oriented sentiment analysis of Russian messages posted by Twitter users. The messages are about banks and telecommunication companies.

The goal of the task is detection of sentiment (negative, neutral or positive) with respect to organizations (banks or telecommunication companies) mentioned in Twitter message. Thus it can be viewed as three-class classification task. The organizers of the evaluation provided labeled training datasets along with unlabeled test datasets for both banks and telecommunication companies. Training datasets contain about 9,000 Twitter messages each, while test datasets contain about 19,000 messages each.

In this paper, we focus on detection of overall sentiment of messages. Object-oriented sentiment classification with algorithms used in this paper is a part of our further research.

All variants of our sentiment analysis system use supervised machine learning algorithms. One of our main goals is evaluation of artificial neural networks (ANNs) for sentiment analysis task. In this paper, we evaluate algorithms based on recurrent neural network (RNN) and convolutional neural network (CNN) along with shallow machine learning approach—SVM with domain adaptation. In each of these three cases we use word2vec (Mikolov et al., 2013a) vectors as features for the algorithms.

We have submitted three solutions to SentiRuEval-2016. The first two are based on recurrent neural network and convolutional neural network, respectively. The last solution is an ensemble solution consisting of three classifiers. It uses SVM with domain adaptation along with RNN and CNN.

The paper is organized as follows: Section 1 provides overview of the related work; Section 2 presents full description of our methods and features that we used; Section 3 provides evaluation results for different methods; in the final section we make conclusion for this work.

1. Related work

Artificial neural networks have become very popular in recent years. They have been shown to achieve state-of-the-art results in various NLP tasks, outperforming shallow machine learning algorithms like support vector machines (SVMs), hidden Markov models and conditional random fields (CRFs).

Recurrent neural networks (RNNs) are considered to be one of the most powerful models for sequence modeling. The success of RNNs in the area of sentence classification was reported by many researchers (Irsoy & Cardie, 2014) (Adamson & Turan, 2015) (Tang et al., 2015).

Convolutional neural networks (CNNs) are another class of neural networks initially designed for image processing. However, CNNs have been shown in recent years to perform very well in NLP tasks, including sentiment analysis and sentence modeling tasks (Kalchbrenner et al., 2014) (Kim, 2014) (dos Santos et al., 2014).

It has been shown that neural network based models for NLP become especially powerful when they are pre-trained with some vector space model (Collobert et al., 2011). The most common way to do this is to use distributed representations of words. The most popular such model now is word2vec (Mikolov et al., 2013a), which improves many NLP tasks.

2. Method description

2.1. Word2vec

Word2vec (Mikolov et al., 2013a) (Mikolov et al., 2013b) is a popular model for computationally efficient learning vector representations of words. Vectors learned using word2vec have been shown to capture semantic information between words (Mikolov et al., 2013c), and pre-training using word2vec leads to major improvements in many NLP tasks.

We used original word2vec toolkit[1] for obtaining vector representations of Russian words. The model was trained on 3.3 GB of user-submitted posts from VK, LiveJournal, echo.msk.ru and svpressa.ru. All the text was lowercased, and punctuation was removed. The following parameters were used for learning:

1. Continuous Bag-of-Words (CBOW) architecture with negative sampling (10 negative samples for every prediction);
2. vector size of 200;
3. maximum context window size of 5;
4. 5 training iterations over corpus;
5. words occurring in the corpus less than 25 times were discarded from the vocabulary; the resulting vocabulary size was 249,014.

2.2. Recurrent neural network

Recurrent neural networks (RNNs) are a class of neural networks that have recurrent connections between units. This makes RNNs well-suited to classify and predict sequence data, including short documents.

Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) is a popular RNN architecture designed to cope with long-term dependency problem. LSTM has been shown to achieve state-of-the-art or comparable to state-of-the-art results in many text sequence processing tasks (Sutskever et al., 2014) (Palangi et al., 2015).

Gated Recurrent Unit (GRU) (Cho et al., 2014) is a simplified version of LSTM that has been shown to outperform LSTM in some tasks (Chung et al., 2014), although according to (Jozefowicz et al., 2015) the gap between LSTM and GRU can often be closed by changing initialization of LSTM cells.

Our RNN-based model is composed of LSTM/GRU network regularized by dropout with probability of 0.5 and succeeded by fully connected layer with 3 neurons that predict probabilities of each class—negative, neutral and positive. The input sample is lowercased and converted to sequence of corresponding word2vec vectors described in section 2.1. Punctuation and words that are not in word2vec vocabulary are discarded. The resulting sequence of vectors is input to the network. Like word2vec vectors, the size of input and output of LSTM/GRU cells is 200.

We tried several variations of recurrent networks: shallow LSTM/GRU, bidirectional GRU and two-layer GRU. We also tried to revert the order of input vector sequences.

We used Keras library[2] to implement the model[3]. In case of LSTM initialization of cells recommended in (Jozefowicz et al., 2015) was used. Sigmoid and hard sigmoid were used for recurrent network as output activation and hidden activation, respectively; softmax was used as activation of fully connected layer.

[1] https://code.google.com/archive/p/word2vec/

[2] https://github.com/fchollet/keras

[3] Source code is available on https://github.com/arkhipenko-ispras/SentiRuEval-2016-RNN

Adam optimizer (Kingma & Ba, 2014) and batch size of 8 were used for training; the number of training epochs was set to 20.

2.3. Convolutional neural network

Due to widely reported success of CNNs (convolutional neural networks) (Kalchbrenner et al., 2014) (Kim, 2014) (dos Santos et al., 2014) in the area of sentiment analysis we have conducted some experiments with CNN as well.

We used word2vec word vectors described in section 2.1 as features. For each tweet the matrix *S* is constructed where s_i (*i*-th row) is a word vector for the *i*-th word in tweet. Then we calculate two vectors t^{avg} and t^{max} as follows:

$$t_j^{avg} = \frac{1}{m} \sum_{1 \le i \le m} s_{ij} \tag{1}$$

$$t_j^{max} = \max_{1 \le i \le m} s_{ij} \tag{2}$$

Concatenation of these two vectors is input to our CNN. The network is composed of convolutional layer with 8 kernels of width 10 which is succeeded by dense layer with 3 neurons (with softmax activation) that predict probabilities of each class. scikit-neuralnetwork library[4] was used for implementing the network. The number of training epochs was set to 10.

The roadmap for further survey includes experiments not only with different kinds of features but also with architecture of the CNN as well. Feature extraction with word2vec seems to be the most promising one. Since CNNs are not as powerful in sequence processing as RNNs the technique of Dynamic k-Max Pooling (Kalchbrenner et al., 2014) can be used to address the problem of variable sentence length.

2.4. Domain adaptation and ensemble solution

2.4.1. Domain adaptation

In most cases we assume that source domain (train data) and target domain (test data) are driven from the same probability distribution:

$$P_s(X, y) \equiv P_t(X, y) \tag{3}$$

Consequently this means that it is impossible to build classifier that would be able to distinguish target domain sample from source domain sample. But in many real world problems assumption (3) does not hold and

[4] https://github.com/aigamedev/scikit-neuralnetwork

$$P_s(X, y) \neq P_t(X, y) \tag{4}$$

How one can detect that $P_s(X, y) \neq P_t(X, y)$?

1. Quality of the model, measured on source domain (e.g. with cross-validation) is much higher than on the target domain. Some participants of SentiRuEval-2015 faced this problem.
2. Consequence of assumption (3) is impossibility to build classifier which can distinguish target domain from source domain. The ability to build such classifier indicates that assumption (3) does not hold. We were able to achieve F1-score on source vs target domain classification above 0.85.

One can improve quality of algorithm in target domain with different method of domain adaptation. Some methods can be found in (Jiang, 2008).

In this work we use a simple method of domain adaptation—sample reweighting. Let $l(x, y, \theta)$ be a loss function. In order to obtain θ we want to minimize following function:

$$L(\theta) = \sum_{x,y \in X \times Y} (x, y, \theta) P_t(x, y) \to \min_{\theta} \tag{5}$$

We can write function L in the equivalent form:

$$L(\theta) = \sum_{x,y \in X \times Y} (x, y, \theta) \frac{P_t(x, y)}{P_s(x, y)} P_s(x, y) \tag{6}$$

Now replace true loss function with empirical estimation:

$$\hat{L}(\theta) = \frac{1}{l} \sum_{i=1}^{l} (x_i, y_i, \theta) \frac{P_t(x_i, y_i)}{P_s(x_i, y_i)} \tag{7}$$

As one can see that algorithm leads as to the feature reweighting with $w_i = \frac{P_t(x_i, y_i)}{P_s(x_i, y_i)}$. Finally we assume that $P_t(y|x) \equiv P_s(y|x)$, thus weight w_i can be found as $w_i = \frac{P(x_i|t)}{P(x_i|s)}$. With Bayes' theorem one can estimate weight as:

$$w_i = \frac{P(t|x_i)P(s)}{P(s|x_i)P(t)} = C \times \frac{P(t|x_i)}{P(s|x_i)} \tag{8}$$

We estimate weight with the logistic regression, and it slightly increases the quality.

2.4.2. Our ensemble solution

Our ensemble classifier consists of three classifiers; each of them votes with equal weight. The first two are GRU neural network and convolutional neural network described in sections 2.2 and 2.3, respectively.

The third classifier is SVM with sample reweighting described in 2.4.1. We used polynomial kernel with degree of 3. For every tweet, the average of word2vec vectors (described in section 2.1) of all words in the tweet is used as features for the SVM classifier.

3. Evaluation

Tables 1–2 present results of the evalution on sentiment classification. Both tables show macro-averaged F1-score of negative and positive classes, used as quality measure on SentiRuEval-2016 competition.

For recurrent neural network based model, we performed 5-fold cross-validation on training data provided by organizers of SentiRuEval. The results are showed in Table 1. We found that GRU network slightly outperforms LSTM network, and that reversing the order of words in tweets improves the quality. Adding an extra recurrent layer also slightly increases the quality.

In addition, we found that using word2vec vectors as features for recurrent network is crucial. Using randomly initialized embedding layer and one-hot features instead of word2vec features gives macro-averaged F1-score of only 0.45 for banks and 0.47 for telecommunication companies.

Table 2 shows results on SentiRuEval test datasets for solutions described in sections 2.2–2.4. It also shows micro-averaged version of F1-score and includes solutions' ranks among all 58 solutions submitted to SentiRuEval by 10 teams. For test data classification with GRU network, the model was trained on whole train data 5 times and correspondingly gave 5 predictions for test data. Then the leading class over all predictions was chosen for each sample. Other models were trained and predicted once.

The Gated Recurrent Unit based solution got the best macro-averaged score on both domains, significantly outperforming solutions from other teams on banks domain, and also has the best micro-averaged F1-score on banks domain.

Table 1. Macro-averaged F1-score, evaluated with RNN models using 5-fold cross-validation on SentiRuEval training data

RNN Architecture	Domain	
	Banks	**Telecommunication companies**
LSTM	0.6026	0.6410
GRU	0.6129	0.6428
GRU, reversed sequences	0.6211	0.6570
Bidirectional GRU	0.6207	0.6521
Two-layer GRU, reversed sequences	0.6243	0.6597

Table 2. F1-score and ranks among all solutions, evaluated on SentiRuEval test data (according to SentiRuEval results)

	Domain			
	Banks		**Telecommunication companies**	
Classifier	**Macro (score/rank)**	**Micro (score/rank)**	**Macro (score/rank)**	**Micro (score/rank)**
CNN	0.4832 / 21	0.5253 / 21	0.4704 / 41	0.6060 / 36
Two-layer GRU, reversed sequences	0.5517 / 1	0.5881 / 1	0.5594 / 1	0.6569 / 21
Ensemble classifier	0.5352 / 2	0.5749 / 2	0.5403 / 9	0.6525 / 23
Best solution not from our team	0.5252 / 3	0.5653 / 3	0.5493 / 2	0.6822 / 1

Conclusion

We have described all variants of our sentiment analysis system. The GRU network based solution performed well and won the SentiRuEval-2016 competition on both domains (banks and telecommunication companies).

Using word2vec vectors as features has made a major contribution to the result. However, we believe that parameters of our classifiers were not optimal, even for GRU network. After publication of labeled test data by organizers of the competition, we were able to achieve macro-averaged F1-score above 0.6 on test data for both domains using GRU network. One of the parts of our future work is to find optimal architectures and learning parameters for RNN and CNN. It is also possible to combine RNN and CNN into one compound network.

In addition, our future research includes adapting our neural network based approaches to object-oriented sentiment analysis, as well as developing methods of domain adaptation within these approaches.

Acknowledgements

This work was supported by the Russian Foundation for Basic Research grant 15-37-20375.

References

1. *Adamson A., Turan V. D.,* (2015), Opinion Tagging Using Deep Recurrent Nets with GRUs, available at: https://cs224d.stanford.edu/reports/AdamsonAlex.pdf
2. *Cho K., van Merrienboer B., Gulcehre C., Bougares F., Schwenk H., Bengio Y.,* (2014), Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, CoRR, available at: http://arxiv.org/abs/1406.1078
3. *Chung J., Gulcehre C., Cho K., Bengio Y.,* (2014), Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, CoRR, available at: http://arxiv.org/abs/1412.3555
4. *Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P.,* (2011), Natural language processing (almost) from scratch, CoRR, available at: http://arxiv.org/abs/1103.0398
5. *dos Santos C., Maira G.,* (2014), Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts, Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, pp. 69–78
6. *Graves A.,* (2012), Supervised Sequence Labelling with Recurrent Neural Networks, available at: http://dx.doi.org/10.1007/978-3-642-24797-2
7. *Hochreiter S., Schmidhuber J.,* (1997), Long Short-Term Memory, Neural computation, volume 9, number 8, pp. 1735–1780
8. *Irsoy O., Cardie C.,* (2014), Opinion Mining with Deep Recurrent Neural Networks, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 720–728
9. *Jiang J.,* (2008), Domain Adaptation in Natural Language Processing, available at: http://hdl.handle.net/2142/11465
10. *Jozefowicz R., Zaremba W., Sutskever I.,* (2015), An Empirical Exploration of Recurrent Network Architectures, Proceedings of the 32nd International Conference on Machine Learning (ICML-15), Lille, France, pp. 2342–2350
11. *Kalchbrenner N., Grefenstette E., Blunsom P.,* (2014), A Convolutional Neural Network for Modelling Sentences, CoRR, available at: http://arxiv.org/abs/1404.2188
12. *Kim Y.,* (2014), Convolutional Neural Networks for Sentence Classification, CoRR, available at: http://arxiv.org/abs/1408.5882
13. *Kingma D. P., Ba J.,* (2014), Adam: A Method for Stochastic Optimization, CoRR, available at: http://arxiv.org/abs/1412.6980
14. *Mikolov T., Chen K., Corrado G., Dean J.,* (2013a), Efficient Estimation of Word Representations in Vector Space, CoRR, available at: http://arxiv.org/abs/1301.3781
15. *Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.,* (2013b), Distributed Representations of Words and Phrases and Their Compositionality, CoRR, available at: http://arxiv.org/abs/1310.4546
16. *Mikolov T., Yih W., Zweig G.,* (2013c), Linguistic Regularities in Continuous Space Word Representations, Proceedings of NAACL HLT 2013, Atlanta, USA, pp. 746–751

17. *Palangi H., Deng L., Shen Y., Gao J., He X., Chen J., Song X., Ward R. K.,* (2015), Deep Sentence Embedding Using the Long Short Term Memory Network: Analysis and Application to Information Retrieval, CoRR, available at: http://arxiv.org/abs/1502.06922
18. *Sutskever I., Vinyals O., Le Q. V.,* (2014), Sequence to Sequence Learning with Neural Networks, CoRR, available at: http://arxiv.org/abs/1409.3215
19. *Tang D., Qin B., Liu T.,* (2015), Document Modeling with Gated Recurrent Neural Network for Sentiment Classification, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 1422–1432