Sentiment Analysis Using Deep Learning

Nikolay Karpov, Alexander Lyashuk and Arsenii Vizgunov

Abstract The study was aimed to analyze advantages of the Deep Learning methods over other baseline machine learning methods using sentiment analysis task in Twitter. All the techniques were evaluated using a set of English tweets with classification on a five-point ordinal scale provided by SemEval-2017 organizers. For the implementation, we used two open-source Python libraries. The results and conclusions of the study are discussed.

Keywords Natural language processing · Sentiment analysis · Deep learning

1 Introduction

There is currently a growing interest in social network analysis due to the expanded role of the latter. The fundamental trend is that people are mostly communicating and collaborating inside these social networks. According to the statistics, the trend of the popularity of social networks is growing steadily. The majority of people now have at least one account at a social network and people communicate there via exchanging text messages. It seems to be extremely important to be able to analyze this type of data and reveal its hidden properties.

One of the most popular formats of communication in a social network is a phenomenon of a post. It is an arbitrary message, expressing thoughts and ideas of a speaking person. Such a post usually appeals to peoples emotions and audience start to actively discuss it by putting more and more comments. An increasing amount of publications in the sphere of the Internet texts analysis reveals quite interesting

N. Karpov (✉) · A. Lyashuk · A. Vizgunov
Higher School of Economics, National Research University, Moscow, Russia
e-mail: nkarpov@hse.ru

A. Lyashuk
e-mail: lx.lyashuk@gmail.com

A. Vizgunov
e-mail: anvizgunov@hse.ru

V. A. Kalyagin et al. (eds.), *Computational Aspects and Applications in Large-Scale Networks*, Springer Proceedings in Mathematics & Statistics 247,
https://doi.org/10.1007/978-3-319-96247-4_20

properties of modern texts. In particular, one can assign sentiment index to every piece of a text. The task usually involves detecting whether it expresses a POSITIVE, a NEGATIVE, or a NEUTRAL sentiment.

While sentiment analysis seems to stay popular, more and more mathematical models are applied to automated prediction of the sentiment. We investigate new models mentioned in literature in comparison with other baseline methods.

The remaining part of the current study is the following. In Sect. 2, we will give the detailed overview of the existing approaches and significant theoretical trends. Further, in Sect. 3, we will describe the experiment methodology we elaborated to perform experiments. The detailed overview of the proposed approach and baseline methods will be given in the Sect. 4. Results of the proposed approach will be shown in the Sect. 5. Finally, we will make a conclusion and further research directions analysis in the Sect. 6.

2 Literature Review

Our research is based on the modern research for automatic natural language processing. The task to automatically identify a sentiment polarity is usually formulated as a classification problem. A classification problem of short messages is a well-known problem in the natural language processing field. This problem is traditionally solved by using machine learning approaches. For instance, sentences can be classified according to their readability, using pre-built features and classification algorithms like SVM, Random Forest, and others [5].

A relevant task for sentiment analysis is explored by many researchers [7, 9]. The practical applications of this task are wide, from monitoring popular events (e.g., Presidential debates, Oscars, etc.) to extracting trading signals by monitoring tweets about public companies.

If quantification of the distribution of sentiment towards a topic across a number of tweets is more interesting than the polarity of each message, this task is usually called quantification [4, 6]. For instance, distribution of sentiment of USA social network users during a particular period is helpful to predict future movements of stock market indicators like DJIA (Dow Jones Industrial Average) or S&P500 [10].

The competition platform for sentiment analysis in Twitter has been run since 2013 and is called SemEval [11]. These applications often benefit greatly from the best possible accuracy, which is why the SemEval-2017 Twitter competition promotes research in this area.

During last few years, neural networks become very popular for various machine learning tasks. Recent advances in Deep Learning allow us to effectively analyze user sentiment handling a big number of messages in social networks. [1] Two of the most popular deep-learning techniques for sentiment analysis are CNNs and LSTMs [2].

In the next section, we show our empirical results of applying deep-learning method in contrast to some traditional machine learning methods.

Table 1 Source data table

	Id	Topic	Text	p
0	628949369883000832	@microsoft	dear @Microsoft the newOoffice...	−1
1	628976607420645377	@microsoft	@Microsoft how about you make...	−2
2	629023169169518592	@microsoft	I may be ignorant on this issue...	−1
3	629179223232479232	@microsoft	Thanks to @microsoft, I just may...	−1
4	629186282179153920	@microsoft	If I make a game as a #windows1...	0

3 Experiment Setup

We consider message polarity classification task. It is required to define sentiment conveyed in the statement from Twitter on a five-point scale from negative to positive. We use SemEval-2017 dataset that contains English tweets. Each sentence is placed with the label (−2 strongly negative, −1 negative, 0 neutral, 1 positive, 2 strongly positive). For evaluation, we use macro-averaged mean absolute error (MAE). For all implementations, we use open-source Python libraries.

3.1 Dataset Overview

We have 10,000 training and 20,000 test objects. Each object contains a tweet id, subject, text, and polarity (Table 1).

3.2 Data Preprocessing

We removed id and topic features and made values of polarity from 0 to 4 instead of −2 to 2 (Table 2).

Table 2 The source data used for training

	Text	p
0	dear @Microsoft the newOoffice...	1
1	@Microsoft how about you make...	0
2	I may be ignorant on this issue...	1
3	Thanks to @microsoft, I just may...	1
4	If I make a game as a #windows1...	2

Table 3 The preprocessed data used for training

	Text	p
0	Dear the newoffice for mac is great and...	1
1	How about you make system that doesn eat...	0
2	May be ignorant on this issue but should we...	1
3	Thanks to just may be switching over to...	1
4	If make game as universal app will owners...	2

When building Machine Learning systems based on tweet data, a preprocessing is required. We used tweet-preprocessor that makes it easy to clean tweets from URLs, hashtags, mentions, reserved words, etc. After that, the data is ready to be used (Table 3).

4 Proposed Approach

4.1 Baseline

First, we transformed tweets into TF-IDF terms and used baseline methods:

- Logistic Regression
- Decision Tree
- Gradient Boosting
- K Nearest Neighbors
- Multilayer Perceptron
- Naive Bayes
- Random Forest
- Support Vector Machine

It can be seen from the bar chart that the perceptron showed the best results. Therefore, we decided that we need to move towards neural networks (Fig. 1).

4.2 Deep Learning

We used Recurrent Neural Network (RNN) with Long Short-Term Memory layers (LSTM) [3]. Neural networks cannot work with a text directly. It needs to work with numerical features. We did not apply the same features as it was for baseline methods. Taking a word sequence in consideration, we transformed tweets into numerical sequences of the same length (Fig. 2 and Table 4).

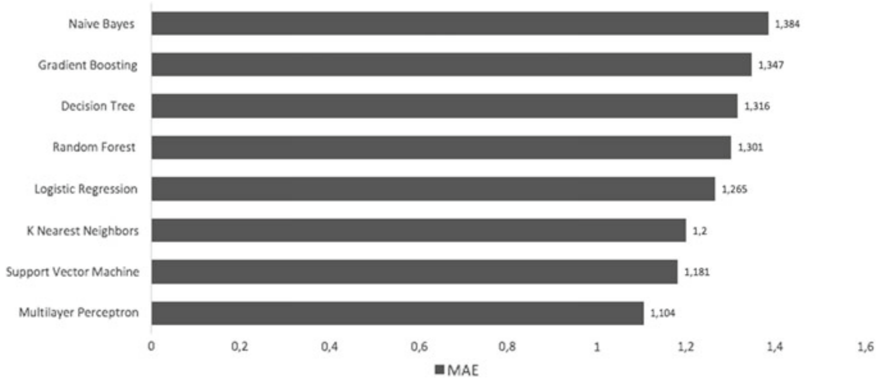

Fig. 1 Amount of the MAE for baseline methods

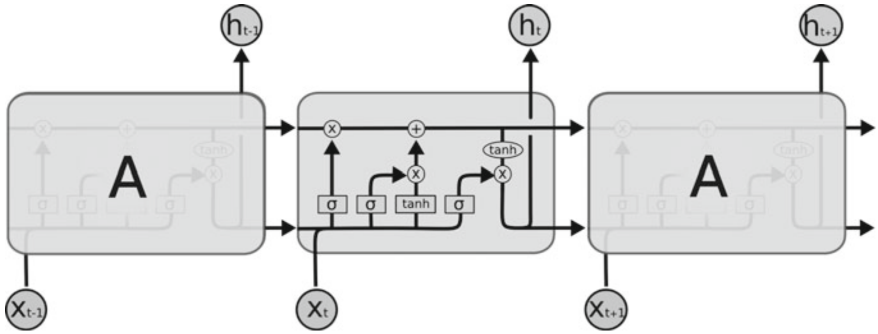

Fig. 2 Structure of the LSTM layer

Table 4 The preprocessed data used for training in neural network

	Text	p
0	[1182, 1, 13480, 8, 715, 7, 132, 4, 32, 21, 66...	1
1	[88, 40, 10, 77, 1998, 17, 286, 1229, 20, 9673...	0
2	[11, 13, 3280, 5, 22, 949, 21, 124, 33, 615, 9...	1
3	[385, 2, 23, 11, 13, 6623, 110, 2]	1
4	[29, 77, 87, 39, 4675, 942, 26, 3659, 13, 738...	2

Inside the neural network, we used a vector representation of words. This is a special representation of a word in an n-dimensional space. Each word corresponds to a vector of n real numbers. The idea is that if the words are similar, then they are located side by side. Here, for example, how it would look in two-dimensional space (Fig. 3).

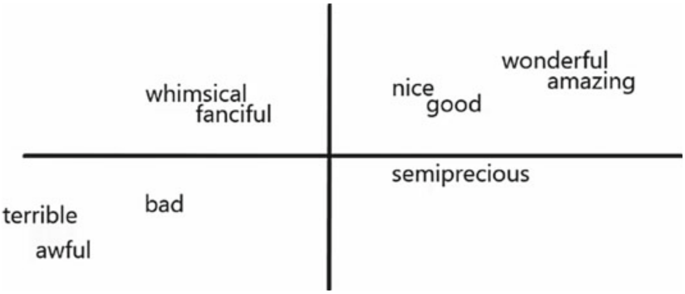

Fig. 3 Two-dimensional example of vector representation of words

To implement the neural network, we used the Keras library,[1] which, in fact, is the shell for the Theano library in our case.

We use a sequential model to input a sequence of words (the layers go one by one).

1. The Embedding layer—the layer for the vector representation of words. The settings indicate that there are 25,000 different words in the dictionary, the sequence consists of no more than 50 words, and the dimension of the vector representation is 200.
2. Two LSTM layers.
3. The Dropout layer is responsible for overfitting. It nulls the random half of the features and prevents the coadaptation of the weights in the layers.
4. Dense—fully connected layer.
5. Activation layer—gives an integer value from 0 to 4 (using the softmax function of activating).

Listing 1.1 Neural Network Definition.

```
max_features = 25000
maxlen = 50
embedding_dim = 200

model = Sequential()
model.add(Embedding(max_features, embedding_dim,
                    input_length=maxlen))
model.add(LSTM(64, return_sequences=True))
model.add(LSTM(64))
model.add(Dropout(0.5))
model.add(Dense(5))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics = ['accuracy'])
```

[1] https://keras.io.

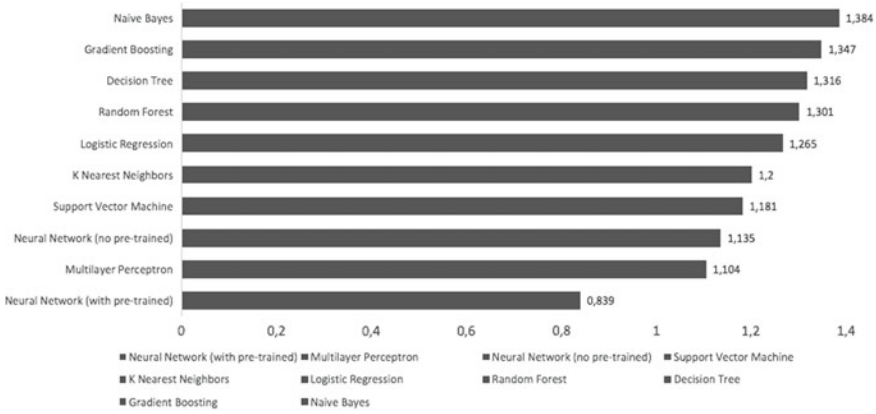

Fig. 4 Amount of MAE for evaluated method with baselines

First, we tried the approach with an automatic selection of the weights of the Embedding layer. But after that, we tried to use Global Vectors for Word Representation (GloVe) [8]. We used pretrained word vectors trained on 2 billion tweets as Embedding layer.[2]

5 Experimental Results

Neural networks show the best result of message polarity classification. A recurrent neural network with a long short-term memory and pretrained Embedding layer of the vector representation of words showed the best MAE indicator—0.8390. The reason why this neural network is better than the network without using the pretrained Embedding layer of the vector representation of words is most likely that the size of the training sample is quite small—10,000 tweets and the set is very noisy. Consequently, the use of the pretrained Embedding layer of the vector representation of words compensates for the small size of the training set and improves the quality of the classification, due to the fact that the pretrained word vectors are tailored to the language model of the social network Twitter. Our open repository includes all source codes and experimental results[3] (Fig. 4).

6 Conclusion

The aim of this research was to apply a simple deep-learning approach to predict sentiment of users which they express in messages of a social network. This approach

[2] https://nlp.stanford.edu/projects/glove/.

[3] https://github.com/lxdv/SemEval-2017.

is compared with some baseline method. We have used a dataset prepared by organizers of SemEval-2017 shared task, five points scale subtask. Using this dataset we have built some machine learning models to automatically predict a polarity of an arbitrary message from any social network user. Experimental results showed that model based on LSTM neural network with pretrained word embeddings allow to obtain significantly better results. We explain this by the fact that neural network with pretrained word embeddings used more data, even in an unsupervised regime.

Acknowledgements The reported study was funded by RFBR according to the research Project No 16-06-00184 A.

References

1. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: EMNLP, pp. 740–750 (2014)
2. Cliche, M.: Bb_twtr at SemEval-2017 task 4: twitter sentiment analysis with CNNs and LSTMs. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 564–571, Vancouver, Canada. Association for Computational Linguistics, Aug 2017
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
4. Karpov, N.: NRU-HSE at SemEval-2017 task 4: tweet quantification using deep learning architecture. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 681–686, Vancouver, Canada. Association for Computational Linguistics, Aug 2017
5. Karpov, N., Baranova, J., Vitugin, F.: Single-sentence readability prediction in Russian. In: International Conference on Analysis of Images, Social Networks and Texts_x000D_, pp. 91–100. Springer (2014)
6. Karpov, N., Porshnev, A., Rudakov, K.: NRU-HSE at SemEval-2016 task 4: comparative analysis of two iterative methods using quantification library. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 171–177, San Diego, California. Association for Computational Linguistics, June 2016. bibtex: karpov-porshnev-rudakov:2016:SemEval
7. Kiritchenko, S., Mohammad, S.M., Salameh, M.: SemEval-2016 task 7: determining sentiment intensity of English and Arabic phrases. In: Proceedings of SemEval, pp. 42–51 (2016)
8. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, vol. 14, pp. 1532–1543 (2014)
9. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B.: Orphée de clercq, véronique hoste, marianna apidianaki, xavier tannier, natalia loukachevitch, evgeny kotelnikov, nuria bel, salud marıa jiménez-zafra, and gülsen eryigit. 2016. SemEval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval, vol. 16 (2016)
10. Porshnev, A., Redkin, I., Karpov, N.: Modelling movement of stock market indexes with data from emoticons of twitter users. In: Russian Summer School in Information Retrieval, pp. 297–306. Springer (2014)
11. Rosenthal, S., Farra, N., Nakov, P.: SemEval-2017 task 4: sentiment analysis in twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 493–509, Vancouver, Canada. Association for Computational Linguistics, Aug 2017