Application of the BERT Language Model for Sentiment Analysis of Social Network Posts

Chapter · September 2020

DOI: 10.1007/978-3-030-59535-7_20

3 authors, including:

Vadim Moshkin
Ulyanovsk State Technical University

Nadejda Yarushkina
Ulyanovsk State Technical University

Application of the BERT language model for sentiment analysis of social network posts

Vadim Moshkin[1[0000-0002-9258-4909]] and Andrey Konstantinov[1[0000-0002-6018-0133]] and Nadezhda Yarushkina[1[0000-0002-5718-8732]]

[1] Ulyanovsk State Technical University, Ulyanovsk, Russia, 432027, Severny Venetz str.,32
{v.moshkin, a.konstantinov, jng}@ulstu.ru

Abstract. The paper proposes an original algorithm for the training dataset formation for a neural network that provides sentiment analysis of social networks posts. This article also describes the use of a neural network to determine the sentiment values of a social network posts using the word2vec and BERT algorithms. Also conducted experiments confirming the effectiveness of the proposed approaches.

Keywords: sentiment analysis, BERT, word2vec, neural network, social network.

1 Introduction

The study of social networks every year is becoming increasingly important because of the need to ensure the safety of the population and the monitoring of public sentiment. Post analysis can help assess changes in the mood of many users and find application in political and social studies including consumer preference research.

The results of the sentiment analysis of the user posts allow us to conclude:

- emotional evaluation of users of various events and objects;
- individual user preferences;
- some features of the users' nature [1].

Sentiment text analysis is a classification task. At present, the best results of text classification by several criteria are shown by machine learning algorithms. Hence, when using neural network approaches, the training dataset formation is relevant.

In this paper, we consider the use of the word2vec and "BERT" language models for sentiment analysis of social networks posts and the solution of the tasks of preprocessing text data and generating a training datasets.

2 The use of machine learning algorithms in sentiment analysis of social network data

Currently, researchers suggest the use of neural networks of various architectures to determine the sentiment values of texts. The support vector machine (SVM) [2], Bayesian models [3], various kinds of regressions [4], methods Word2Vec, Doc2Vec [5], CRF [6], as well as convolutional and recurrent neural networks [7] [8].

More steps are required for preprocessing resources including for the training set formation for sentiment analysis of social networks texts. This distinguishes them from the analysis of formal and strictly structured texts.

The paper [9] describes a sentiment analysis model for posts from the social network Twitter. Initially, a set of smiles was created for marking up the text and assigning the text to a specific emotion. Then the texts were presented in vector form using the "bag of words" approach.

Three classifiers were chosen to construct the classification model: logistic regression, decision tree, multilayer perceptron. The accuracy of determining the sentiment values of the posts was about 75-76% for each model.

In [10], 2 models of neural networks were selected to determine the sentiment values of text messages: a neural network with two recurrent layers and a neural network with recurrent and convolutional layers.

The authors used two sets of hand-labeled texts for training a neural network. Texts are short messages up to 140 characters long. The classification accuracy was 69% using a network with two recurrent layers. The accuracy is slightly higher - 71% using a network with recurrent and convolutional layers.

The paper [11] presents the results of the development of an automatic classifier of Russian-language Internet texts. This classifier distributes texts into 8 classes in accordance with 8 basic emotions.

The classifier was based on the SVM. The input values for the classifier are various linguistic parameters, for example, the frequency of use of punctuation marks and amplification adverbs. The accuracy of determining the emotional coloring of emotions "anger" and "fear" was 48%, "anguish" - 40%, "disgust" - 6%, and "joy" - 7%.

As can be seen from the results of the above studies, the task of developing an approach to effectively assess the sentiment values of social networks texts is relevant.

3 An approach to sentiment analysis of social network data using the "word2vec" and "BERT" models.

An approach was developed to sentiment analyze the text data of social networks in the framework of this study. This approach includes the following steps:

1. The formation of training and test sets.
2. Text vectorization using the word2vec and BERT models.
3. Training and classification using the neural network approach.

1.1 The Algorithm for the formation of the training set

The formation of training and test sets requires pre-processing of textual information and the marking of the sentiment values of individual text posts.

Formally, the process of selecting posts can be represented by the scheme shown in Figure 1. Each stage of the selection shown in the figure includes the process of selecting posts for each specific emotion.

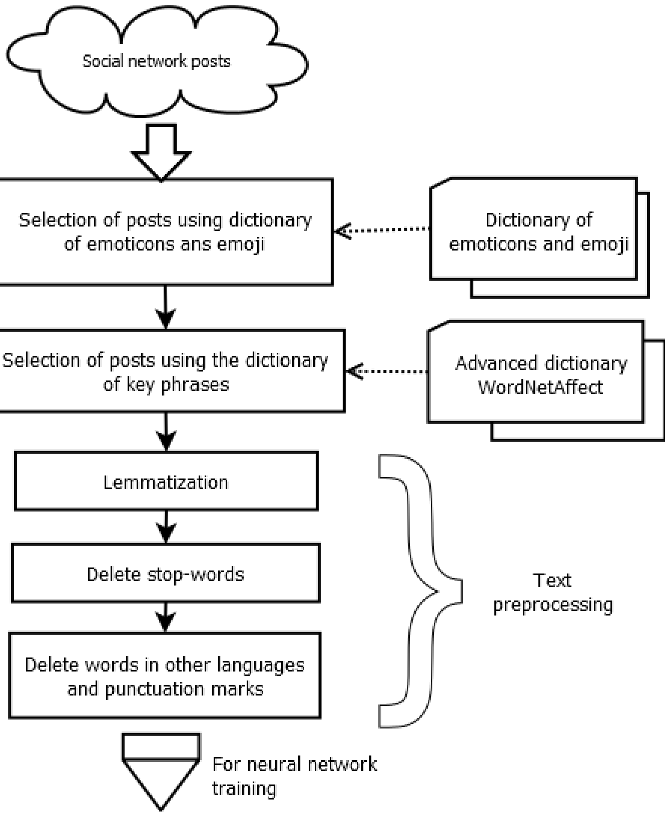

Fig. 1. Posts selection.

1. At the first stage, posts are selected based on expert dictionaries of emotion expression symbols (the so-called "emoticons" and "emoji"). If a post contains an author's symbol for expressing emotions, then it belongs to a specific class and is added to the corresponding list.

2. The second stage is the selection of posts based on dictionaries of key phrases. An extended Russian-language semantic thesaurus WordNetAffect [12] was used as a basic dictionary.

The developed dictionaries with symbols of expression of emotions and key phrases consist of objects of 7 classes:

$$D^E = \{D^E_{joy}, D^E_{sad}, D^E_{surp}, D^E_{anger}, D^E_{disg}, D^E_{cont}, D^E_{fear}\}$$

where D^E_{joy} is a class of objects with emotion "joy", D^E_{sad} is a class of objects with emotion "sadness", D^E_{surp} is a class of objects with emotion "surprise", D^E_{anger} is a class of objects with emotion "anger", D^E_{disg} is a class of objects with emotion "disgust", D^E_{cont} is a class of objects with emotion "contempt", D^E_{fear} is a class of objects with emotion "fear".

In addition, at this stage, the lemmatization of each word of the post is performed. Then the post is checked for the content of each word from the dictionary. If a post contains a phrase, then it belongs to a specific class of emotional coloring.

3. At the stage of preprocessing posts, all characters are excluded except for Cyrillic characters and spaces, and all words are reduced to lower case.

1.2 Text vectorization algorithms

Two methods were used to represent words in a vector space: word2vec and "BERT" in the framework of this study.

The model of the BERT algorithm can be represented as a function, the input of which is text, and the output is a vector. In this algorithm, each syllable is converted to a number. Initially, a model trained for a particular language is loaded, according to which the sequence is divided into syllables. A detailed description of the algorithm is given in [13] and [14].

The mathematical model of the BERT algorithm. The loaded model can be represented as:

$$\Theta = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ w_n \end{bmatrix},$$

where Θ is a long vector that contains the words w included in the dictionary of words of the loaded model. The algorithm converts a word into a set of syllables or vectors, each syllable is obtained from a set of common words.

Let w_1, w_2 ... w_n be the set of words in the dictionary and s_{m1}, s_{m2} ... s_{mn} the set of syllables in the word w_n, then the function $(s_m) = f(w_{11}, w_{12} \dots w_{1n})$ allows you to get many syllables for a sequence of words.

Then we get a vector representation of the sequence of words by the resulting syllables.

The word2vec algorithm was also used for comparison. The word2vec algorithm converts words to vectors. A detailed description of the algorithm is given in [15].

The mathematical model of the word2vec algorithm. Initially, a dictionary of all the words that make up the dataset is compiled. Formally, all word vectors are:

$$\Theta = \begin{bmatrix} V_{w1} \\ V_{w2} \\ . \\ V_{wn} \\ U_{w1} \\ U_{w2} \\ . \\ U_{wn} \end{bmatrix},$$

where Θ is a long vector that contains vectors *v* and *u* of length *d* for all words.

The algorithm predicts the probability of a word in its context. Vector vectors are obtained; each word is assigned a probability value that is close to the probability value of meeting a word in this environment in real text.

$$P(w_o|w_c) = \frac{e^{s(w_o,w_c)}}{\sum w_i \in V^{e^{s(w_o,w_c)}}}$$

where w_o is the vector of the target word, w_c is some context vector calculated (for example, by averaging) from the vectors surrounding other words of the desired word. And $s(w_1,w_2)$ is a function that maps one number to two vectors.

Word probabilities are predicted and optimized in the standard model discussed above. The function for optimization is the Kullback–Leibler divergence:

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

where *p (x)* is the probability distribution of words that is taken from the dataset, *q (x)* is the distribution that is generated by the model. Divergence is how much one distribution is not like the other.

1.3 Neural network model for sentiment analysis of text fragments

A neural network model can be represented in the form of layers used in its architecture. A neural network consists of seven layers and is shown in Figure 2.

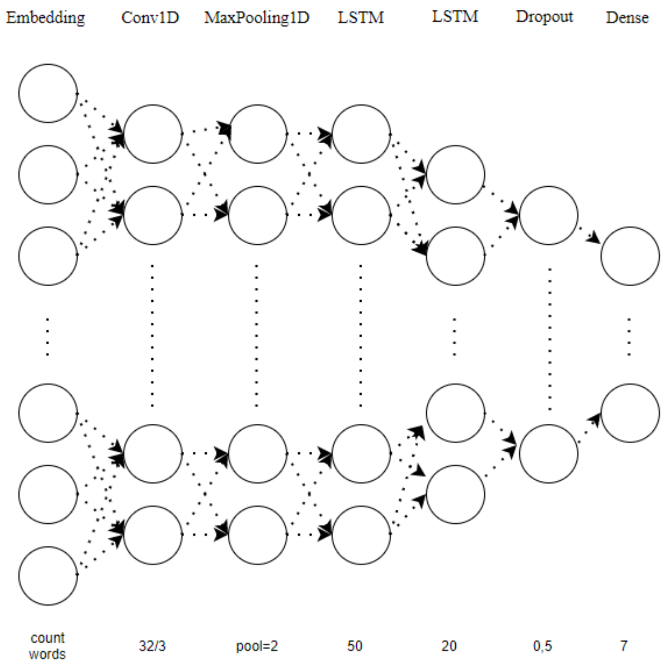

Fig. 2. The architecture of the developed neural network.

Mathematically, a neuron is a weighted adder, the only output of which is determined through its inputs and a weight matrix as follows:

$$y = f(u),\ where\ u = \sum_{i=1}^{n} w_i x_i + w_0 x_0$$

where x_i and w_i are the signals at the inputs of the neuron and the weight of the inputs, respectively. The function u is called the induced local field, and *f (u)* is called the transfer function. Possible values of the signals at the inputs of the neuron are considered set in the interval [0,1]. The additional input x_0 and the corresponding weight w_0 are used to initialize the neuron.

Initialization refers to a shift in the activation function of a neuron along the horizontal axis.

Each neuron is associated with the concept of an activation function, which can be summarized as:

$$f(x) = tx$$

where *t* is the factor responsible for the distribution of the activation function. The proposed neural network architecture has the following set of layers:

• The Embedding layer is the input layer of the neural network:

$$Emb = \{Size(D), Size(S_{vec}), L_{Sec}\},$$

where $Size(D)$ is the size of the dictionary in the text data, $Size(S_{vec})$ is the size of the vector space into which the words will be inserted, $Size(S_{vec}) = 32$, L_{Sec} is the length of the input sequences equal to the maximum size of the vector formed during word processing.

- The Conv1D layer is a convolutional layer, necessary for deep learning. With this layer, the accuracy of the classification of posts is increased by 5-7%. The number of filters is 32, each filter has a length of 3. The activation function is “ReLU”.
- Layer MaxPooling1D - a layer responsible for storing temporary data. The maximum pool is 2.
- The LSTM layer is a recurrent neural network layer. The model uses 2 LSTM layers, one consists of 50 neurons, the second consists of 20 neurons.
- The Dropout layer is needed to avoid retraining the neural network. A value of 0.5 is given as a parameter, which means that a neural network can exclude up to half of inactive neurons.
- The Dense layer is an output layer of seven neurons. Each neuron is responsible for a specific emotion.

4 Software implementation and experimental results

3.1 An example of the training dataset formation

Consider an example that demonstrates how the algorithm for selecting posts for obtaining a training sample works. Take 7 posts, which are presented in table 1.

Table 1. The first stage of the selection of posts.

Text	Emotion
i love summer 😀	joy
July and August came out very productive -	-
I killed two pairs of sneakers in a year 😟	sad
that’s why it’s so dirty with us	-
photographs reflect well the internal state	-
closing season went excellent 🙂	joy
my body continues to rebel 😫	anger

After the first stage (see chapter 1.1), 7 groups of posts are obtained. Each group contains posts selected based on emotion symbols. The “joy” group contains 2 posts, since they contain copyright symbols of emotions from the “joy” group in the example. The “sadness” and “anger” groups contain 1 post each, since they contain symbols of emotions from the corresponding groups. The remaining groups do not contain posts, since posts containing the author’s emotion symbols for these groups did not meet.

At the second stage, the selection is based on key phrases. Each emotional group is specified based on a dictionary of key phrases. The selection result is presented in table 2.

Table 2. The second stage of the selection of posts.

Text	Emotion
i love summer 😀	joy
I killed two pairs of sneakers in a year ☹	sad
closing season went excellent 😍	joy
my body continues to rebel 😫	anger

The output is posts that contain keywords from the dictionary. For a group, “joy” is the words “love” and “excellent”. For other posts, keywords for this emotional group were not found.

The number of posts after each selection stage is shown in table 3. 2.5 million posts are sent to the entrance. The number of posts in each group is reduced on average by 2-3 times after each stage of selection. It should be noted that the generated set contains posts of various lengths.

Table 3. Formation of the training set.

Emotion	The number of posts after stage 1	The number of posts after stage 2
Joy	237837	74309
Sadness	7274	2629
Surprise	2739	1535
Fear	4640	2436
Anger	1363	512
Contempt	9960	5613
Disgust	5011	1206

3.2 The algorithm for the training dataset formation

A software system of sentiment analysis of social network posts was implemented to evaluate the effectiveness of the proposed models and algorithms.

The neural network was implemented in Python using the TensorFlow and Keras frameworks intended for machine learning. Python was chosen as the programming language.

Input data is 2.5 million posts from the social network VKontakte. Posts were automatically downloaded from open social network groups via the VKontakte API, and contain only textual information [16].

The neural network, consisting of seven layers, was trained at a different number of posts, from 500 to 1000 and above. The number of learning eras is 100. The set was divided into training and test, 90% and 10%, respectively.

Posts were selected whose length was in a certain interval (40-50 words or 290-310 characters) for experiments 4-7. Otherwise, because of the many short posts in which the vector is supplemented with zeros, the neural network is not trained.

During the experiments, the hypothesis was verified that the training set, formed on the basis of the author's emoticons and key phrases, is better than:

- a set formed only on the basis of key phrases.

- a set formed only on the basis of emoticons.

3 training sets for experiments were formed:

• based on emoticons and key phrases (includes the first and second stages of selecting posts);

• based on only emoticons (includes only the first stage of the selection of posts, the second stage of selection is excluded);

• based on only key phrases (includes only the second stage of selection of posts, the first stage of selection is excluded).

The results of the experiments are shown in table 4.

Table 4. The results of the experiments.

№	Algorithm	Number of posts	Selection training set by	The dataset is balanced	Class weights	Post length	Accuracy in the training set	Accuracy on test dataset
1	word2vec	1042	emoticons and keywords	no	no	40-50 words	0,98	0,77
2	word2vec	1042	emoticons and keywords	no	yes	40-50 words	0,97	0,79
3	BERT	556	emoticons and keywords	no	no	290-310 characters	0,95	0,86
4	BERT	556	emoticons and keywords	no	yes	290-310 characters	0,95	0,87
5	BERT	726	emoticons	no	yes	290-310 characters	0,94	0,8
6	BERT	726	emoticons	no	no	290-310 characters	0,91	0,82
7	BERT	2100	keywords	yes	no	290-310 characters	0,87	0,83
8	BERT	513	emoticons and key-words, no stop-words	no	yes	290-310 characters	0,95	0,82

The experiments show that the highest accuracy when using the BERT algorithm, class weights are set, since the sample is unbalanced - the accuracy is 0.87.

Experiments №4-6 show that a sample formed on the basis of copyright symbols of emotions and key phrases is better than a sample only on the basis of copyright symbols of expression of emotions or only based on key phrases.

Experiment №8 shows that a neural network trained on a sample with stop words has higher accuracy than a trained on a sample without stop words.

Conclusion

As a result of the work, the neural network LSTM architecture was used to determine the emotional coloring of the posts of the social network. The best result was when using the BERT algorithm for text processing. During the study, an accuracy rate of determining the emotional coloring of the posts in 87% was achieved.

In future studies, it is planned to improve the algorithm for the formation of the training sample, including by expanding the dictionaries used by automating the process of their formation.

Acknowledgement. This study was supported by the Russian Foundation for Basic Research (Grants No. 18-47-732007, 18-47-730035 and 19-07-00999).

References

1. Vlasov, D. et al. Description of the information image of a user of a social network taking into account its psychological characteristics // International Journal of Open Information Technologies. - 2018. - T. 6. - No. 4.
2. Sabuj, M.S., Afrin, Z., Hasan, K.M.A. (2017) Opinion Mining Using Vector Machine for Web Based Diverse Data. / Pattern Recognition and Machine Intelligence. PReMI 2017. Lecture Notes in Computer Science, vol 10597. Springer, pp. 673-678.
3. Dinu, L.P., Iuga, I. (2012) The Best Feature of the Set. In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2012. Lecture Notes in Computer Science, vol 7181. Springer, pp. 556-567.
4. Chetviorkin, I.I., Loukachevitch, N.V. Sentiment Analysis Track at ROMIP-2012. Computational linguistics and intellectual technologies. Computer Linguistics and Intelligent Technologies: Dialogue 2013. Sat scientific articles volume 2, pp. 40-50.
5. Qufei, C. and Sokolova, M. “Word2Vec and Doc2Vec in Unsupervised Sentiment Analysis of Clinical Discharge Summaries.” CoRR abs / 1805.00352 (2018): n. pag.
6. Antonova, A., Soloviev, A. Using the conditional random field method for processing texts in Russian. Computer Linguistics and Intelligent Technologies: Dialogue 2013. Sat scientific articles / Vol. 12 (19) - M.: Publishing House of the Russian State Humanitarian University, 2013.– pp.27-44.
7. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. The International Language Technologies-Volume 1 International Association for Computational Linguistics, pp. 142-150.
8. Moshkin V., Yarushkina N., Andreev I. The Sentiment Analysis of Unstructured Social Network Data Using the Extended Ontology SentiWordNet, IEEE, 12th International

Conference on Developments in eSystems Engineering (DeSE), Kazan, Russia, 2019, pp. 576-580. 10.1109/DeSE.2019.00110.
9. Bogdanov, A. L., Dulya, I. S. Sentiment analysis of short Russian-language texts in social media // Bulletin of Tomsk State University. Economy. 2019. №47. pp. 159-168
10. Smirnova, O.S., Shishkov, V.V. The choice of neural network topology and their application for the classification of short texts // International Journal of Open Information Technologies. 2016. No 8. pp. 50-54.
11. Kolmogorova, A.V., Vdovina L.A. Lexico-grammatical markers of emotions as parameters for sentiment analysis of Russian-language Internet texts // Bulletin of Perm University. Russian and foreign philology. 2019. №3. pp. 38-46.
12. WordNetAffect , http://wndomains.fbk.eu/wnaffect.html, , last accessed 2020/04/21.
13. Devlin, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding // arXiv preprint arXiv: 1810.04805. - 2018.
14. Horev, R. BERT Explained: State of the art language model for NLP, https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270, last accessed 2020/04/21.
15. Algorithm Word2Vec, https://neurohive.io/ru/, last accessed 2020/04/21.
16. Filippov A., Moshkin V., Yarushkina N. (2019) Development of a Software for the Semantic Analysis of Social Media Content. In: Dolinina O., Brovko A., Pechenkin V., Lvov A., Zhmud V., Kreinovich V. (eds) Recent Research in Control Engineering and Decision Making. ICIT 2019. Studies in Systems, Decision and Control, vol 199. Springer, Cham pp. 421-432