Textual Analysis of News for Stock Market Prediction

Alexander V. Bogdanov, Maxim Bogan, and Alexey Stankus(✉)

Saint Petersburg State University, 7-9 Universitetskaya emb., Saint Petersburg 199034, Russia
alexey@stankus.ru

Abstract. Stock market prediction constitutes an important factor in business. There are a large number of different mathematical models for predicting stock price movements. One of the alternative approaches is application of methods based on Natural Language Processing (NLP).

Though NLP tasks are getting popular, they remain complex and voluminous. In the digital age almost, all information has been transferred to digital records that is good achievement. That is a good achievement. But on the other side because of the ease in creating new information, information search in Internet becomes complicated. This problem becomes more relevant, and scientists continue to search ways to structure a huge amount of information.

There are many methods of traditional representations of words based on statistics. But these methods don't give representation about contexts and semantics of text document. In this paper, we will consider approaches that help to get semantics from news. To evaluate our methods, we will use them for predicting direction of S&P 500 Index. In other words, we will compare our approaches with a stock market prediction problem based on news.

Keywords: Recurrent neural network · Stock prediction · LSTM · Glove · Word2Vec · News · Word Embedding · Tf-idf · YAKE

1 Introduction

There are a number of works aimed at predicting the movement of stock prices based on news [1–5]. But in these works, news headlines used as input data. News contains full information about the event, and the information in the headline may not reflect the whole essence of the news. At the same time, these works do not cover in detail the issue of preliminary processing of texts as input data, but only the analysis of a neural network.

Our idea is to predict the direction of price movement based on the content of the entire text of the news. Taking the news texts themselves as an input parameter leads us to the task of a deeper analysis of the news texts themselves and determining the best way to prepare the data before using it in machine learning models.

News is a set of `n` sentences, consisting of `m` words, where `n` and `m` are different for each news item. Initially, we need to understand what data format we need to pass on

O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12956, pp. 313–323, 2021.
https://doi.org/10.1007/978-3-030-87010-2_22

to the machine learning algorithms in order to work correctly. Every machine learning algorithm must have a constant number of input variables in the processed data. Therefore, any preprocessing method must create a fixed vector of variable length of news texts.

One of the key questions is how to get a fixed number of functions from a different number of news objects. This issue will be discussed in this article. In this article, we will check several data preparation algorithms for both classifiers (Fig. 1) and neural networks (Fig. 2) and compare the work of each of them.

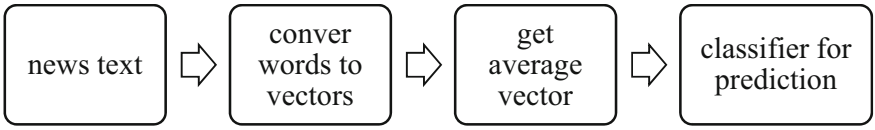

Fig. 1. Work with classifier diagram.

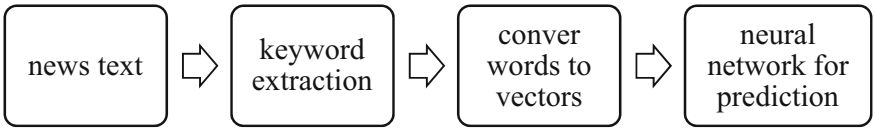

Fig. 2. Work with neural network diagram.

2 Work Vector Representation Model

In this section, we will first introduce methods for vector representation of words. After that, we will consider approaches to obtaining a news vector from vector words, which will correspond to some average meaning of the whole news.

2.1 Bag of Words Model and Word Embedding

There are two main models of vector representation of words - Bag of Words [6] and Word Embedding [7]:

The Bag of Words model creates a representation of a document as an unordered collection of words with no knowledge of the relationships between them. This algorithm creates a matrix, each row of which corresponds to a separate document or text, and each column corresponds to a specific word. Table 1 shows that the cell at the intersection of the row and column contains the number of occurrences of the word in the corresponding document. The main disadvantages of this approach are the lack of semantic meaning of words and entire documents, and does not consider the word order, which plays a big role. Therefore, this approach is not suitable for the task set in this work.

The Word Embedding model brings together a variety of natural language processing approaches. In this model, a corresponding fixed-length vector in an n-dimensional

Table 1. Word distribution in texts.

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc1	1	1	1							
Doc2	1		1	1	1	1				
Doc3					1	1	1	2	1	1

vector space is constructed for each word. But it is built in such a way as to maximize the semantic connection between words. One of the ways to express the semantic connection of words in a vector space is cosine similarity. It is calculated according to the following formula (1):

$$cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}, \quad (1)$$
$$where\ A_i\ and\ B_i - is\ the\ i - th\ element\ of\ vectors\ A\ and\ B$$

Within the Word Embedding model, the following algorithms have been developed:

- Word2Vec [8]
- GloVe [9].

In the field of NLP, these models are the most modern and in demand today. Therefore, to create a vector representation of words that contains the semantic meaning of words, we will use the Word2Vec and Glove models.

2.2 Preparation of Texts

At this step, we have a vector representation of each word in the news corpus. A word represents a vector in n-dimensional vector space, that is, a word is a set of n - features, note that n is fixed. But in every news, the number of words is always different. Therefore, it is necessary to bring each news to a fixed number of signs.

Simple Averaging
The first way is simple - it is to take all the vector words belonging to the news and calculate the average vector, following formula (2):

$$\overrightarrow{\omega_{average}} = \frac{\sum_{i=1}^{m} \overrightarrow{\omega_i}}{m}, \quad (2)$$
$$where\ \omega_i\ vector\ of\ the\ i - th\ word,\ m - the\ number\ of\ words\ in\ the\ news.$$

Thus, we get a vector corresponding to the whole news. And if the vector of each word reflects the semantic meaning, then the vector of the whole news reflects some average meaning of the news. As a result, each news item is matched with a vector of dimension n, where n is fixed.

For example, the news: “Shanghai stocks opened lower, and the yuan was weaker against the dollar on Wednesday. Other Asian markets also declined. Hong Kong’s

benchmark declined 2% as the city faced heightened tensions". Converted to a single vector as follows:

$$\left.\begin{array}{c} \text{Shanghai} = \overrightarrow{(0.345, 0.101, \ldots, 0.640)} \\ \text{stocks} = \overrightarrow{(0.783, 0.089, \ldots, 0.554)} \\ \cdots = \ldots \\ \text{heightened} = \overrightarrow{(0.421, 0.484, \ldots, 0.054)} \\ \text{tensions} = \overrightarrow{(0.383, 0.211, \ldots, 0.954)} \end{array}\right\} \rightarrow \overrightarrow{\omega_{average}} = \overrightarrow{(0.497, 0.301, \ldots, 0.628)}$$

Term Frequency - Inverse Document Frequency (tf-idf)

Words in the news have different meanings or importance. Then it is worth not just counting the average, but constructing a linear combination, where each vector will be multiplied by a coefficient corresponding to the importance of the word. This idea is contained in the tf-idf (term frequency - inverse document frequency) formula [10]. It considers the frequency with which the word occurs in the text, a weighted average is taken.

tf - the ratio of the number of occurrences of a certain word to the total number of words of one text document.

$$tf(t, d) = \frac{n_t}{\sum_k n_k}, \tag{3}$$

где n_t– is the number of times the word t has been mentioned in the document, and the denominator is the total number of words in the document.

idf is the inverse of the frequency of occurrence of a word in all documents. Accounting for idf reduces the weight (weight) of commonly used words.

$$idf(t, D) = log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}, \tag{4}$$

where $|D|$ – number of documents, $|\{d_i \in D | t \in d_i\}|$ – the number of documents in which we have t(when $n_t \neq 0$).

Hence, the tf-idf measure is the product of two factors:

$$tfi{-}df(t, d, D) = tf(t, d) \times idf(t, D) \tag{5}$$

Words with a high frequency within a particular document and with a low frequency of use in other documents will receive a lot of weight in tf-idf.

So, using the tf-idf measure, a linear combination of vectors is built and divided by the number of vectors. As a result, we get a vector of fixed length, which is also some mean sense of the news.

The example from "Simple averaging" the transformation looks like formula (6):

$$\left.\begin{array}{c}\overrightarrow{(0.345, 0.101, \ldots, 0.640)} \cdot 0.76 \\ \overrightarrow{(0.783, 0.089, \ldots, 0.554)} \cdot 0.34 \\ \cdots = \ldots \\ \overrightarrow{(0.421, 0.484, \ldots, 0.054)} \cdot 0.56 \\ \overrightarrow{(0.383, 0.211, \ldots, 0.954)} \cdot 0.48\end{array}\right\} \rightarrow \overrightarrow{\omega_{average}} = \overrightarrow{(0.385, 0.094, \ldots, 0.670)} \quad (6)$$

Key Words

Certainly, when calculating the average meaning, the loss of information is possible since the meaning turns out to be very approximate and there are many words that do not carry meaning. Let's try to remove words that do not carry meaning.

We extract a fixed number of keywords from each news item. Further, to transfer several words to the algorithm sequentially, we will use a recurrent neural network LSTM (Long short-term memory). It is a recurrent neural network that can store values for both short and long periods of time.

The YAKE algorithm [11] developed by a team of French scientists [12] in 2018 was chosen to extract keywords. YAKE is an unsupervised automatic keyword extraction method that relies on the statistical characteristics of the text. Moreover, not only nominal and non-nominal entities are extracted in the form of words and noun phrases, but also predicates, adjectives and other parts of speech that carry key information of the news. In [12], the method is compared with ten modern unsupervised approaches such as tf-idf, KP-Miner, RAKE, TextRank, SingleRank, ExpandRank, TopicRank, TopicalPageRank, PositionRank and MultipartiteRank), and one supervised method (KEA). According to the results of the authors of the article, the YAKE method shows the best result.

In each news, using the YAKE algorithm, all keywords are marked in each sentence. Further, all unmarked words are deleted and as a result, only significant words remain. After that, using Word2Vec or Glove, all words are translated into the corresponding vectors, and news, consisting of a sequence of vectors, is fed into the LSTM network, where growth/fall markers (1/0) are at the output. Figure 3 depicts a network model with inner LSTM layers and a convolutional layer.

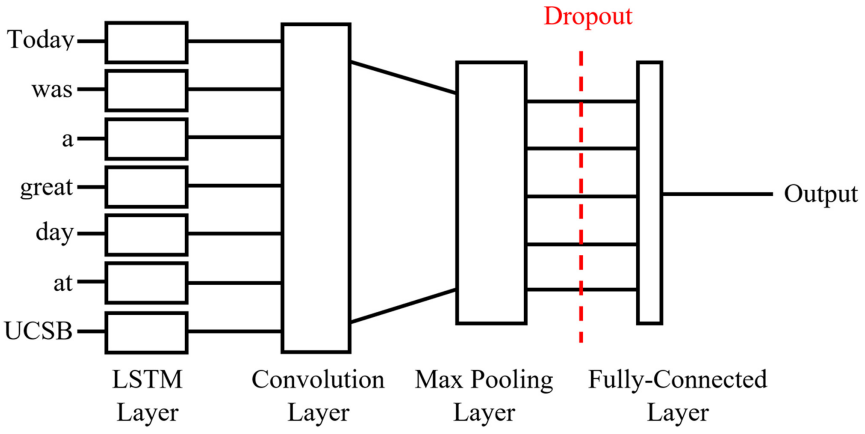

Fig. 3. Model with inner layers.

Let's give an example of processing news using keywords:

"Shanghai stocks opened lower and the yuan was weaker against the dollar on Wednesday. Other Asian markets also declined. Hong Kong's benchmark declined 2% as the city faced heightened tensions".

Therefore, we get:

"Shanghai stocks opened lower yuan weaker dollar Wednesday. Asian markets declined. Hong Kong's benchmark declined city faced heightened tensions".

3 Experiments

3.1 Data

The experiments were conducted on the basis of a set of news collected from BBC News, Breitbart News, CNN, The New York Times, Reuters, Washington Post, Bloomberg and Yahoo News for the period from November 2018 to August 2019 (Table 2). Next, data were taken with prices S&P500 Index. We use the closing price as the price.

Table 2. Time intervals and amount of news.

S&P 500 Index prediction data			
Data set	Training	Development	Testing
Time interval	25/11/2018–21/07/2019	21/07/2019–20/08/2019	20/08/2019–19/09/2019
News	182,250	22,782	22,782

We use the news content published over the course of an hour to predict the S&P 500 movement up or down, comparing the closing price at $t + 1$ with the closing price at t, where t is the trading hour.

3.2 Implementation Details

For Word2Vec and GloVe algorithms, pre-trained word-vector dictionaries exist. So, for Word2Vec there are pre-trained word-vectors on Google news with a dimension of 300. There is a dictionary for GloVe trained on various textual data, including on news with Common Crawl with a dimension of 300. Based on pre-trained word vectors, we will train our word vectors, and for comparison of results we will take the same dimension 300.

To evaluate news processing approaches, we will use classifiers as machine learning algorithms:

- Extra trees
- Support Vector Classifier (SVC) with rbf
- Random Forest
- Logistic regression
- Linear SVC
- Naive Bayes
- Multilayer perceptron.

And also, for keywords - the recurrent neural network LSTM.

As an estimate, we will use the accuracy of the coincidence of the growth and fall of the true and predicted values.

3.3 Prediction Algorithms

Both Word2Vec and GloVe have two subsamples of experiments - these are eigenvectors of words and pre-trained word vectors. Word2Vec and GloVe methods are used in all experiments.

Next, the Averaging and tf-idf methods form a fixed vector. This vector is fed into classifiers and one standard deep learning model:

- Extra trees
- SVC with rbf
- Random Forest
- Logistic regression
- Linear SVC
- Naive Bayes.
- Multilayer perceptron.

But for the keywords method, only the LSTM recurrent neural network is suitable. Because the above classifiers and the standard deep learning model do not have such properties as LSTM - to accept as input a sequence of the same type of feature vectors and extract new features from them. In each model, the best hyperparameters are selected and cross-validation is carried out.

3.4 Results

The results of comparing news preparation approaches to machine learning algorithms in Tables 3 and 4 show that pre-trained vectors give a more accurate vector representation of words. This is due to the fact that the amount of news is probably not enough to determine the exact context of words in a vector representation. A comparison between the approaches of simple averaging and weighted averaging with tf-idf shows that simple averaging gives an idea of a certain mean sense worse than weighted averaging with the tf-idf measure. The vector representation algorithms for Word2Vec and GloVe work similarly, but GloVe in the end gives a 0.5% better result.

Table 3. Prediction results with Word2Vec preprocessing.

Word2Vec						
Embedding	Own trained word vectors			Pretrained word vectors		
Preprocess methods	Average	tf-idf	Key words	Average	tf-idf	Key words
ExtraTrees	54.53%	54.98%	–	56.44%	57.12%	–
SVC with rbf	55.45%	55.58%	–	56.90%	**57.20%**	–
Random forest	55.58%	55.61%	–	55.74%	55.90%	–
Logistic regression	53.02%	54.27%	–	53.23%	54.79%	–
Linear SVC	53.01%	53.98%	–	54.97%	53.56%	–
Naïve Bayes	49.87%	48.79%	–	50.43%	51.86%	–
Multilayer perceptron	53.11%	53.20%	–	53.22%	53.63%	–
LSTM	–	–	56,32%	–	–	**57,63%**

Table 4. Prediction results with GloVe preprocessing.

GloVe						
Embedding	Own trained word vectors			Pretrained word vectors		
Preprocess methods	Average	tf-idf	Key words	Average	tf-idf	Key words
ExtraTrees	55.21%	55.70%	–	57.44%	57.89%	–
SVC with rbf	56.11%	56.23%	–	57.56%	**58.00%**	–
Random forest	56.08%	56.07%	–	56.08%	56.23%	–
Logistic regression	53.32%	54.76%	–	53.61%	55.22%	–
Linear SVC	53.23%	54.31%	–	55.33%	53.97%	–
Naïve Bayes	49.87%	48.79%	–	50.43%	51.86%	–
Multilayer perceptron	53.31%	53.45%	–	53.52%	53.69%	–
LSTM	–	–	57,42%	–	–	**58,97%**

During the experiments in the method based on keywords, the optimal number of keywords was chosen equal to 1900. The experimental data are presented in Fig. 4. But it is worth noting that on average there are 2700 words in the news, which indicates that the selected number is approximately 2/3 of all words. As a result, the accuracy slightly exceeded the maximum value of averaging methods.

Further, a series of experiments was carried out with the dimension parameter of vectors in the case of training our vectors, since the maximum dimension of pre-trained vectors proposed by the authors of the algorithms does not exceed 300. Figures 5 and 6 show schedules of changes in prediction accuracy with respect to the dimension of word vectors.

It can be noted that in both cases, when the dimension reaches 1900, the accuracy of the prediction ceases to grow, and in the future it even worsens. As a result, by increasing the dimension of the vectors, it was possible to increase the accuracy by 0.5–1%.

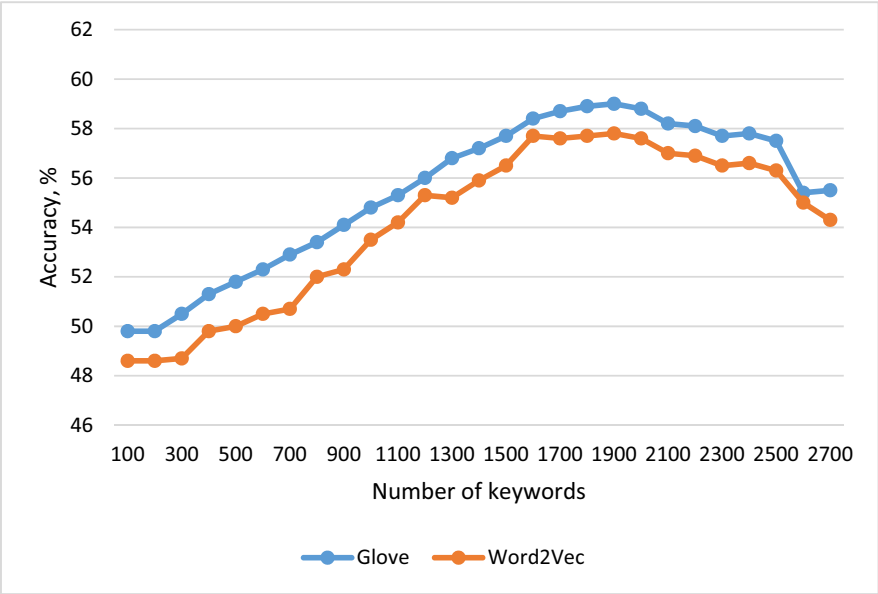

Fig. 4. Optimal number of keywords.

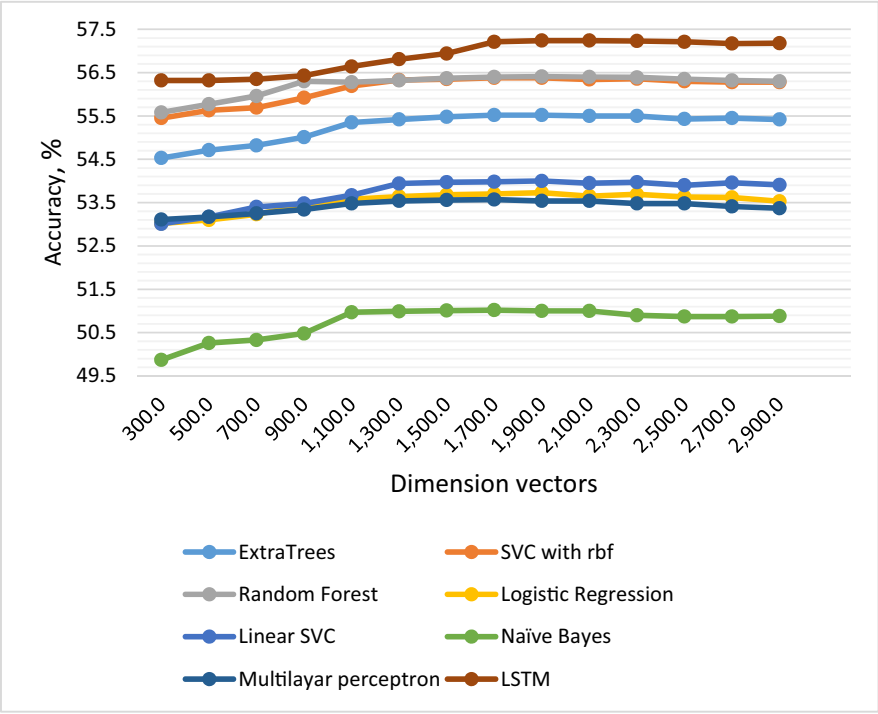

Fig. 5. Prediction changes in Word2Vec

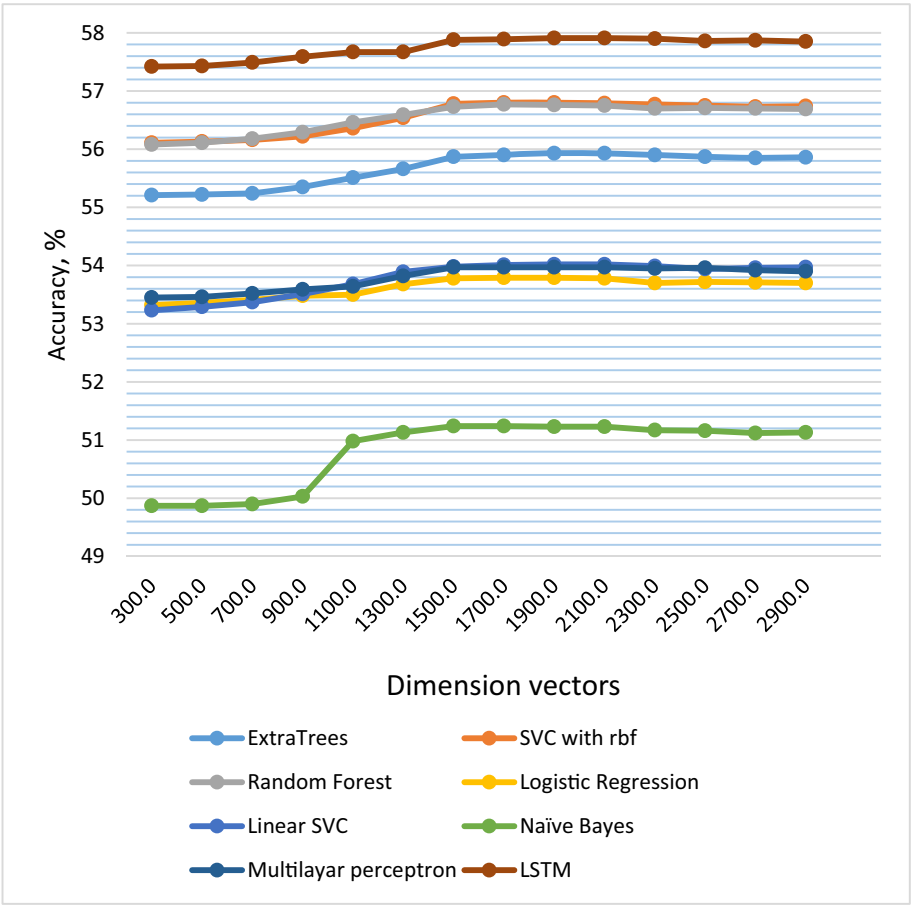

Fig. 6. Prediction changes in Glove.

4 Conclusion

In this paper, we consider news processing methods for feeding into machine learning algorithms. Algorithms based on vector averaging are easy to process, but they can lead to loss of information, such as the word order in the news.

The algorithm based on the extraction of keywords, that is combination of the most important words and LSTM showed the best result (57.63% with Word2Vec and 58.97% with GloVe) and has the prospect of development in such tasks.

To improve the result, a deeper analysis of the texts is necessary, including the determination of the positive or negative meaning of the news. We also believe that using the new Hierarchal Attention Network model can improve news prediction.

References

1. Liu, H.: Leveraging financial news for stock trend prediction with attention-based recurrent neural network (2018)
2. Liu, J., Chao, F., Lin, Y., Lin, C.: Stock prices prediction using deep learning models (2015)
3. Alam, F., Kumar, A., Vela, A.: Using news articles to predict stock movements based on different forecasting techniques statistical, Regression and Text Mining (2018)
4. Zhu, X., Nahar, S.: Predicting stock price trends using news headlines (2016)
5. Velay, M, Daniel, F.: Using NLP on news headlines to predict index trends (2018)
6. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: a statistical framework. Int. J. Mach. Learn. Cybern. (2010)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. ICLR (2013)
9. Pennington, J., Socher, R., Manning. C.D.: GloVe: global vectors for word representation (2014)
10. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. MCB University (2006)
11. Campos R., Mangaravite V., Pasquali A., Jorge A.M., Nunes C., Jatowt A.: YAKE! collection-independent automatic keyword extractor. In: Pasi G., Piwowarski B., Azzopardi L., Hanbury A. (eds.) Advances in Information Retrieval. ECIR 2018. Lecture Notes in Computer Science, vol 10772. Springer, Cham (2018)
12. Campos R., Mangaravite V., Pasquali A., Jorge A.M., Nunes C., Jatowt A.: A text feature based automatic keyword extraction method for single documents. In: Pasi G., Piwowarski B., Azzopardi L., Hanbury A. (eds.) Advances in Information Retrieval. ECIR 2018. Lecture Notes in Computer Science, vol 10772. Springer, Cham (2018)
13. "Finam". https://www.finam.ru/
14. Pre-trained vocabulary Word2Vec. https://code.google.com/archive/p/word2vec/
15. Pre-trained vocabulary GloVe. https://nlp.stanford.edu/projects/glove/