HARVARD EDUCATIONAL AND SCIENTIFIC REVIEW

International Agency for Development of Culture, Education and Science

Harvard Educational and Scientific Review
International Agency for Development of Culture, Education and Science
United Kingdom
Street: 2 High Street City: Ashby Phone number 079 6425 7122
Zip code DN16 8UZ Country United Kingdom
USA
Soldiers Field Boston, MA 02163 +1.800.427.5577

International Agency for Development of Culture, Education and Science United Kingdom
USA Soldiers Field Boston

MODELS AND ALGORITHMS FOR PROCESSING TEXT DATA IN SOCIAL NETWORKS

A.Sh.Karaxanova
Phd student, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Email: shazizovna@gmail.com

Abstract. The article explores different algorithms for clustering large volumes of textual data. Existing implementation methods were analyzed and Word2Vec and GloVe algorithms were selected. The initial text data for testing the algorithms was obtained by collecting notes from groups created through Telegram. The obtained results showed that the use of these algorithms allows to estimate the frequency of use and importance of individual words in relation to the context of the studied community. Also, the results of the application of algorithms were compared in the work and a conclusion was drawn about their effectiveness.
Keywords. Social networks, text data, Word2Vec, Glove, FastText.

Introduction

At present, a large amount of data is constantly being created in the world, such as many transactions, data of mobile operators, etc. Social networks are of particular interest because they represent a huge amount of diverse, constantly updated information. Many companies need to analyze social media data to gauge user reactions to their products. In addition, the analysis of this area is used to solve security issues. By collecting and clustering text data from social networks, it is possible to identify the main topics and events discussed by social network users in different cities and countries.

Word2Vec

Word2Vec is basically a predictive embedding model. It mainly uses two types of architecture to create vector representation of words

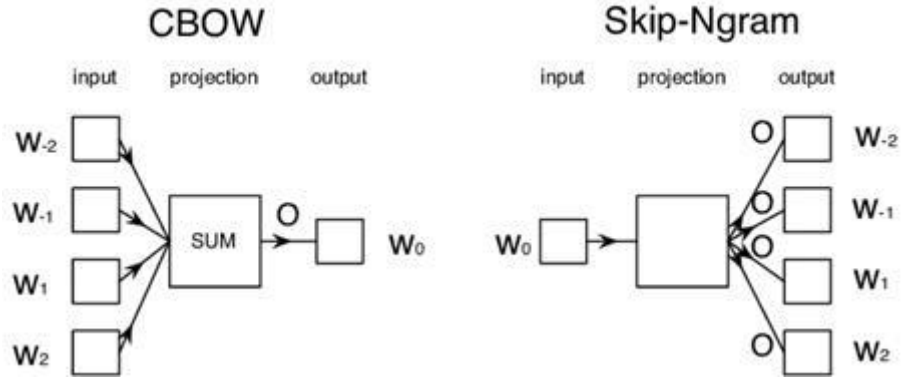
1. Continuous set of words (CBOW)

In this architecture, the model predicts which word is most likely to occur in a given context. Therefore, words with equal probability of occurrence are considered similar words and appear closer in the measurement space .

Suppose we replace the word "boat " with "ship" in a sentence, then the model predicts the probability of both, and if it turns out to be similar, we can consider the words to be similar.

2. Skip-gram

This architecture is similar to CBOW, but instead the model works in reverse. The model predicts the context using a given word.



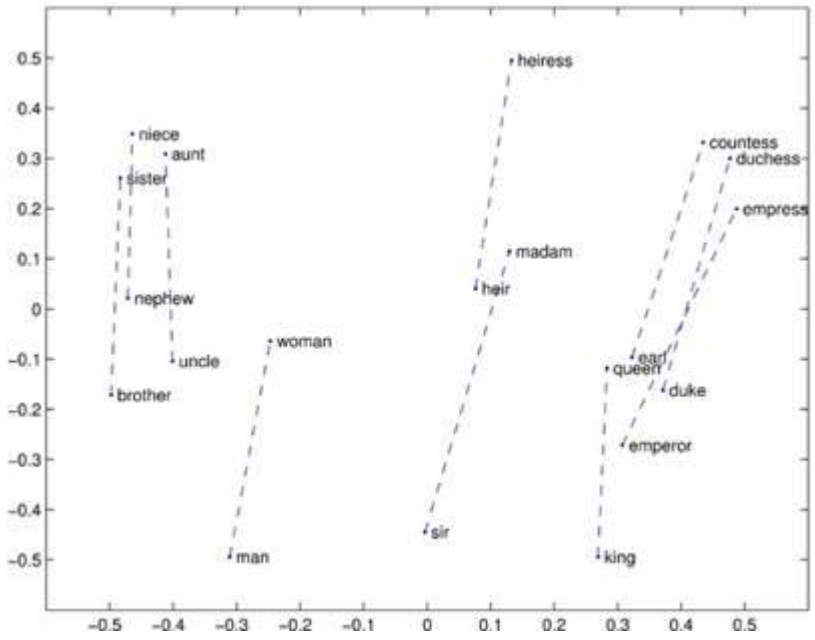
Word2Vec architecture

Clustering allows using common attributes of different classifications to define clusters. By examining one or more attributes or classes, individual data elements can be combined to form a structured summary [2].

Various algorithms are used to perform clustering tasks. One of the most widely used is Word 2 Vec. The idea of the algorithm is not to compare the words themselves or their sequence (called n-grams), but to compare the semantic classes they fall into [3]. a large amount of textual material was obtained (The model was implemented in Python using NumPy, SciPy libraries [4]). The algorithm assigns a vector to each word. Also, adjacent words correspond to adjacent vectors. A measure of the proximity of words is their contextual proximity: nearby words are located next to the same words in the text. The distance between vectors is measured using cosine similarity (cosine similarity) [4-5]. In training neural networks, Word 2 Vec maximizes the cosine proximity measure between vectors of words occurring in similar contexts and minimizes the cosine proximity measure between non-adjacent words. Word 2 Vec outputs the coordinates of the vectors corresponding to the given words. In Word 2 Vec, two different neural network architectures can be used to convert words into vectors: Continuous Bag of Words (CBOW) and Skip-gram. Choosing one of these models is done by specifying the " sg " hyperparameter. By default, sg =0, the CBOW model is used. If sg =1, Skip - gram [7-8] is used. Another hyperparameter is the size of the window in which the context of the given word is considered. This implementation uses a " window " parameter that defines the maximum number of words between a given word and its neighbor in a sentence, words further away from the given word are not counted as its context. Moreover, it does not take into account which word is closer and which word is further away from the given word in the text, if both words fall into the window. Another important hyperparameter " size " is the vector corresponding to the words. is the size. If its value is small, then the model turns out to be rough, but with a large value, the role of machine learning disappears, and matching vector words can become a unitary encoding of words.

GloVe. Both architectures of Word2Vec are predictive, some contextual words occur more often than others, and they only consider the local context and thus fail to capture the global context. The GloVe model is trained on a global word-by-word co-

occurrence matrix from a given text set of text documents. This co-occurring matrix is decomposed to form a denser and more expressive vector image.



Vector image of GloVe

GloVe is another popular unsupervised machine learning algorithm for extracting vector representations of words. The working principle of GloVe is very similar to Word 2 Vec, but unlike Word 2 Vec, which is based on a "prediction" model, GloVe is built on a "calculation" model [9]. It is aimed at solving the problem by getting the meaning of z. First, it traverses the entire corpus and collects statistics on word occurrences, after which it constructs a co-occurrence matrix of words. A word association matrix is a matrix in which each row and column corresponds to a word from the corpus. ri, and at the intersections of rows and columns there are numbers corresponding to the number of words next to the word in the row. the word in the column. The distance at which words should be separated from each other is determined by the algorithm parameter. [10].

The next step is to factorize the matrix. A similar operation occurs in the Word 2 Vec algorithm and is called negative sampling. As a result, GloVe is required to minimize the following functions:

$$J = \Sigma_{i=1}^{V}\Sigma_{j=1}^{V} f\left(\tilde{X}_{ij}\right)_{i}\omega_i^T\omega_j + b_i^{+}b_j - \log(X_{ij}))^2,$$

where V is a dictionary value; ωi is the vector of the key word; ωj is a contextual word vector; b i , bj - scalar shifts; f (Xij) is a weighting function that avoids overfitting in frequently repeated pairs.

A special feature of GloVe is that this algorithm can take into account differences between words that are close enough in the vector plane, but still have certain differences, for example, sibling pairs. For this, GloVe implements a mathematical model that establishes approximately equal vector distances between such words.

FastText

One of the main disadvantages of installing Word2Vec and GloVe is that they cannot encode unknown or non-vocabulary words.
So, to solve this problem, Facebook proposed FastText model. It is an extension of Word2Vec and follows the same Skip-gram and CBOW model. but unlike Word2Vec, which feeds whole words to the neural network, FastText first divides the words into several subwords (or n-grams) and then feeds them into the neural network.
For example, if n is 3 and the word is " apple ", the tri-gram ['<ap', 'app', 'ppl', 'ple', 'le>'] and its so will be entered. This will be the sum of the vector representation of the trigrams. Here, the hyperparameters " minn " and " maxn " are assumed to be 3, and the characters "<" and ">" represent the beginning and end of the word.
Thus, using this methodology, unknown words can be represented in the form of a vector, since its n-grams are more likely to exist with other words.

Results

Similar communities of interest were selected to collect data to analyze the effectiveness of the developed algorithms. We chose Telegram to analyze communities in the social network because it is one of the most popular social networks today . The unique feature of social networking is that it is available to everyone. The next stage of our work was the development of a software module for data collection. The implementation was done in Python using the script library for VKontakte. All interactions with the social network are then done through the module.

After collecting the records of the selected teams, Word 2 Vec implemented in the gensim library was trained on the totality of all texts. For data analysis, we took a matrix with a window width of 5 2 = 10 and a vector size of 100. For each word, a corresponding vector was obtained from the text and the cone similarity between the vectors was calculated. An example of vector distances obtained using World 2 Vec is given in Table 1.

Table 1. The vector distances between the word "music" and other words as words are 2 vec .

Word	Vector distance
For	0.985627
Online	0.751461
Famous	0.721355
to dance	0.691231
new	0.678548

In parallel, the GloVe algorithm was applied to the same data. The implementation of the algorithm involves creating a matrix of vector representations of the form (5,100), where 5 is the maximum number of words and 100 is the size of the representation, each element i is a vector corresponding to the size of the representation includes _ the word with number *i* in the index created during tokenization . Figure 1 shows an example of vector distances obtained using GloVe .

Figure 1. Vector distances between words in the GloVe model.

In order to compare the results of data processing according to the previously described algorithms, 3 main indicators were obtained during the work process: the accuracy of the analysis result, the speed of data processing and the amount of RAM used (Fig. 2).

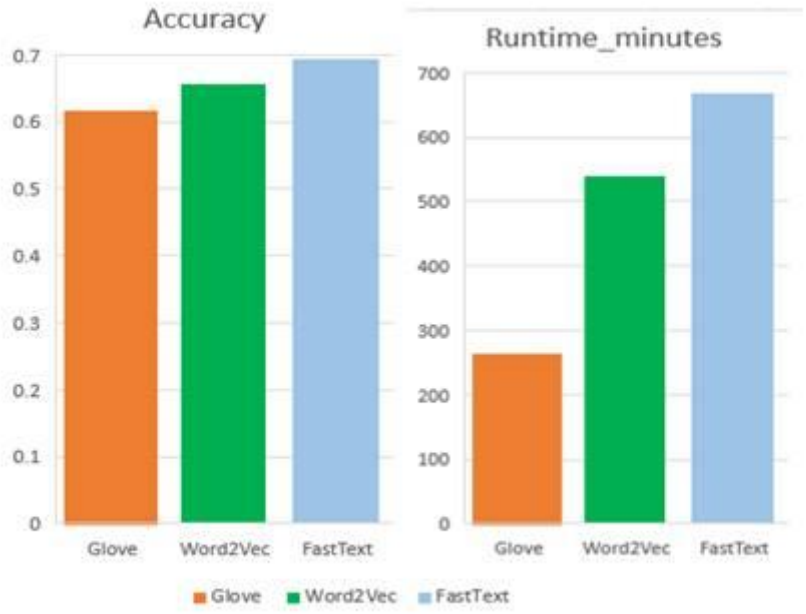


Figure 2. Comparative analysis of the results of using Word 2 Vec and Glove algorithms.

So all three algorithms give the same basic result: a vector for each word, with vectors usefully - relative distances/directions being our general relatedness of words and even certain semantic measures approximately corresponds to our understanding of the dependence on .

Working from the same corpus, generating word vectors of the same size, and paying the same attention to meta-optimization, the quality of their word vectors will

be approximately the same. Significant differences are only in the speed of data processing and the amount of RAM used.

In general, GloVe precomputes a large word exchange matrix in memory and then acts on it on the fly, word 2 vec scans the sentences online and processes each common occurrence separately. So there is a trade-off between using more memory (GloVe) and longer session (word 2 vec). Also, once computed, GloVe can reuse the co-occurrence matrix for fast factorization with any size, while the 2 vec words need to be trained from scratch after resizing the embedding. .

Summary

Problems related to clustering and subsequent classification of text data are relevant due to the worldwide spread of social networks and Internet services. The approaches and methods presented in the article are planned to be tested on text data collected from the Vkontakte social network in the Russian segment. The necessary data is collected using a developed software package. It is planned to develop this topic in the direction of generation and optimization of parallel clustering algorithms.

References

1. Коваленко, Т.В. Разработка библиотеки построения векторной модели текста на основе морфемного разбора слов / Т.В. Коваленко, Р.Б. Галинский, Ю.В. Яковлева, И.В.Никифоров // Неделя науки СПбПУ: материалы научной конференции с международным участием – СПб.: Изд-во Политехн. ун-та, 2017.

2.Vector Representations of Words [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/tutorials/word2vec (21.11.2019).

3. Краткий обзор языка Python [Электронный ресурс]. – Режим доступа: http://www. helloworld.ru/texts/comp/lang/python/python2/index.htm.

4. Mikolov T. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean // Proceedings of NIPS, 2013.

5. Efficient Estimation of Word Representations in Vector Space / T. Mikolov, K. Chen, G. Corrado, J. Dean // Proceedings of Workshop at ICLR, 2013.

6.Rytsarev, I.A. Application of the principal component analysis to detect semantic differences during the content analysis of social network / I.A. Rytsarev, D.D. Kozlov, N.S. Kravtsova, A.V. Kupriyanov, K.S. Liseckiy, S.K. Liseckiy, R.A. Paringer, N.Yu. Samykina // CEUR Workshop Proceedings. – 2018. – Vol. 2212. – P. 262-269.

7.Ю.А. Курбатов, И.А. Рыцарев, А.В. Куприянов Исследование алгоритмов обработки текстовых данных всоциальных сетях// VI Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2020)

8. Global Vectors for Word Representation [Electronic resource]. – Access mode: https:// nlp.stanford.edu/projects/glove/(15.11.2019)..

Vol.3. Issue 3 Pages 41-47.
9.Primqulov O.D. “The role of quantum algorithms in the solution of important problems”, Harvard educational and scientific review, Vol.2. Issue 1 Pg.193-197