Comparative Analysis of Word Embedding Models

Shruti Ahuja, Skandha Choudhry, Dara Nanda Gopala Krishna, Eashan Arora, Vimal Kumar K

Jaypee Institute of Information Technology, Noida, India
shrutiahuja44@gmail.com,skandhachoudhry@gmail.com,nandukanna44@gmail.com,eashanarora2798@gmail.com,vimalkumar.k@gmail.com

Abstract: With the expansion of technology and communications, it has become increasingly important to develop an efficient system that can deal with the huge amounts of raw text data being generated daily. The aim of this paper is to come up with an efficient method to embed words into vectors and process them simultaneously. We have researched upon two models that fall under Word embedding approaches: Continuous Bag of Words (CBOW) model and Skip Gram model. Continuous Bag of Words uses multiple context words to predict the target words while the Skip Gram model uses a single word to make the prediction of multiple target words. We use both these models to process a given dataset and predict the words which are closest and most similar to each other. By comparing both these models side by side, we have determined which one is more efficient and fit for use in real world applications. Our results show that Continuous Bag of Words is more efficient when dealing with an extremely large dataset composed of thousands of words, while Skip Gram has turned out to be more accurate in case of smaller datasets.

Keywords: Continuous Bag of Words. Skip Gram, Word Embedding, Neural Network, Language Model.

1 Introduction

There are various Natural language related applications being developed these days. The aim of these applications ranges from speech enabled assistance to application that helps in the accessing hardware through natural languages. All these applications are in great demand these days due to availability of powerful systems and development of various deep learning models. The natural language related applications need language models in the language being used. The language models need to capture the syntactic and semantic information stored in the text. So, that the accuracy of these applications can be reasonably good. The syntactic and semantic information has to be represented in the form of vectors, so that, it can be used by machine learning models. The process of generating vectors that can hold the syntactic and semantic information of a given text is called as word embedding. As we know, computer processes all of its data in binary form, i.e. in 0s and 1s. All text is converted into binary form for processing and the same context is applied when trying to process large amounts of domain dependent data for the implementation of machine learning and deep learning algorithms. Word Embedding helps in processing

the text through various techniques. In a nutshell, we can say that word embedding is the process of mapping a word to a vector without loss of any information provided in sentence about the word.

There are two major categories of the word embedding processes:

1. **Frequency based Embedding:** This sort of embedding makes use of the frequency of the words that have occurred in the corpus and counting the number of times a particular word is used and embedding the word accordingly. This method involves Count Vector, TF-IDF Vectorization and Co-Occurrence Matrix.
2. **Prediction based Embedding:** This method makes use of neural networks to build a graph of words and predict their closeness(similarity) to each other using contextual information.

In this research paper, we focused on prediction based embedding techniques which are further divided into two categories: Continuous Bag of Words (CBOW) model and Skip-Gram model. Many researchers have worked on word embedding techniques and a summary of work related to this approach is presented in section-2. Two models of prediction-based embedding are further discussed in Section 3. The section-4 analyzes both the models under consideration. The section-5 concludes and presents the future scope of this work.

2 Related Work

Zhong Li Ye et. al. [1] proposed a syntactic word embedding model that can discriminate polysemous words and hold the structural relationship stored in sentences. To discriminate polysemous words, tagging algorithm was proposed using latent dirichlet algorithm (LDA). Dependency based context information are extracted and fed to the proposed embedding model. The syntactic contextual information are extracted using a dependency parser. The author has analyzed this approach and found that it has performance improvement by 20% when compared with existing word embedding models.

Wenhao Zhu et. al. [2] has developed a word embedding model that makes use of pronunciation of word as an additional feature. So that, it can provide an accurate semantic information which can be used in embedding process. This proposed model can be integrated with any existing word embedding model and it can also be used to develop other language models. By the addition of pronunciation, the word embedding model is found to have semantic information which will be helpful in many other applications.

A modification to use the morphological feature was proposed in a skip gram model by Piotr Bojanowski et. al [3]. Each word fed to this model is a character level ngram combinations. The character ngram helps in predicting vectors of words which are not in vocabulary of the language model. The morphological information helps in improving its performance over other word embedding models. Since, the character

level ngram is used for training purpose, the speed of training the model is also increased.

3 Prediction based Embedding Techniques

3.1 Continuous Bag of Words Model

This model considers the neighboring words of a sentence to predict the target word. It simulates the context words over neural network layers by assigning weights to words according to their context and importance. Apart from context and importance, it also uses the semantic similarity between each word. For example,

Text: He goes to fetch water at the _____.

'He', 'goes', 'to', 'fetch', 'water', 'at' and 'the' are taken as input to predict the target word 'river'.

To properly implement this technique, we should first take a raw corpus that can be fed as training set to the model, which can later process it and map words to vectors accordingly. Once the corpus is set, we build the CBOW model and use it to implement Word Embedding.

The prediction model makes use of supervised machine learning for word embedding which can be later used for the purpose of text prediction in various fields. To further explain this, we used the example of 'He is the man. The man is happy. She is the happy woman.' Using this example on Continuous Bag of Words model, we demonstrate that the word most similar to 'man' is 'woman'.

It has also made use of neural networks to calculate the average distance between words based on their context. Similar to a trigram model, the algorithm makes batches of 'He, is, the', 'is, the, man' and so on. After the completion of training, the neural network learns to predict 'woman' as the closest word to 'man'.

3.2 Skip Gram Model

Skip-gram model does the exact opposite of Continuous Bag of Words, in the sense that the Continuous Bag of Words uses multiple context words to predict the target word, whereas, Skip Gram Model predicts multiple target words with the input of only one context word. The neural network is trained with the corpus, and the final loss is then taken into account to predict the nearest words to the context word fed as input. For example,

Text: He_____.

Here, only the word 'he' is given as input and it is used to predict the rest of the words, i.e., 'goes to fetch water at the river'.

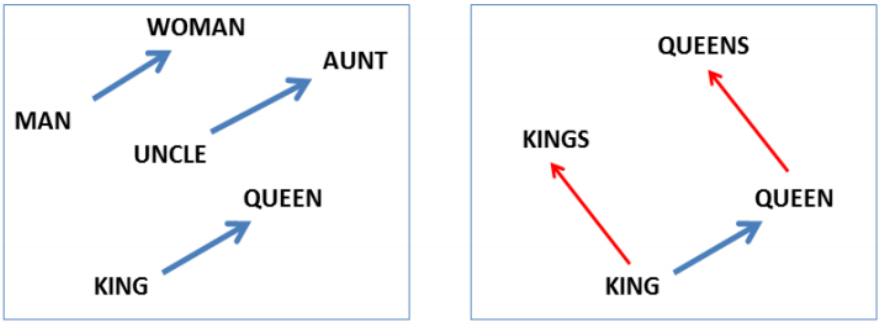

Fig. 1. The basic gist of the idea is to make the computer understand that 'king' is to 'queen', what 'man' is to 'woman' and 'uncle' is to 'aunt'. [4]

Implementing Skip Gram Model would require us to provide a corpus and then build a model which is uses a supervised learning algorithm based on shallow neural networks. The main difference in the approach of the two models is the amount of input fed to predict the target words. There are also other technical differences in the model which will be addressed in next section along with the analysis of both the models.

Towards the end of this report, we would compare CBOW and Skip Gram by providing them with the same raw text. For ease of understanding, we're taking the sentence ' He goes to fetch water at the river' as an example right now by taking this sentence as well 'He goes to fetch water at the well' as the training sets.

Application of Neural Network Layers. We have applied three layers in this network, an input layer, a hidden layer and an output layer. The input layer has all the unique words in the corpus as the training set. The hidden layer makes use of the weights assigned to each word in the corpus to calculate the words closest to each other in terms of nature and similarity. The size of the output layer is same as the input layer.

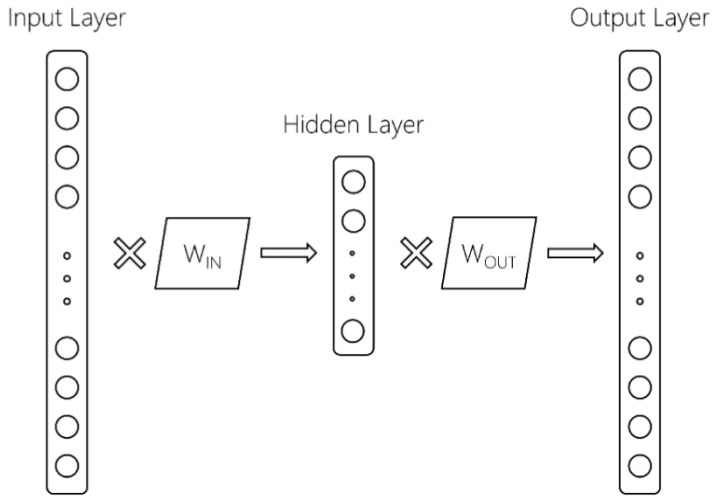

Fig. 2. A model of the neural network layers in the word2vec model.

By calculating the distance of the word with the actual target word, the neural network is trained multiple times to minimise the loss of efficiency that occurs at each epoch. The network then finally arrives at the final step with the minimum loss and it gives ‘ocean’ as the closest word to ‘river’. This mainly occurs because of the creation of the batches ‘well, water’ and ‘river, water’ which goes on to make the computer predict the words ‘well’ and ‘river’ as similar in nature.

4 Analysis

Now, we have used the same raw text to compare both the approaches in terms of efficiency and results. The raw data is fed as input to the network for training the algorithm. The network analyzes data and generates batches of words, which is further embedded into vectors. The batches of words used in this model are in a combination of three consecutive words and it has also been analyzed using unigram, bigram and ngram language model. But it is found to perform well in case of a trigram model as compared with other models. It also optimizes the network performance and reduces the loss. Thus, making the network learn features.

The major difference between the two models is in the working. CBOW uses the average gradient for embedding the words while the Skip Gram model makes use of the normal distribution formula to calculate the nearest words. This is because CBOW uses multiple inputs, whose average is calculated to give the output while Skip Gram takes single input for the generation of multiple outputs.

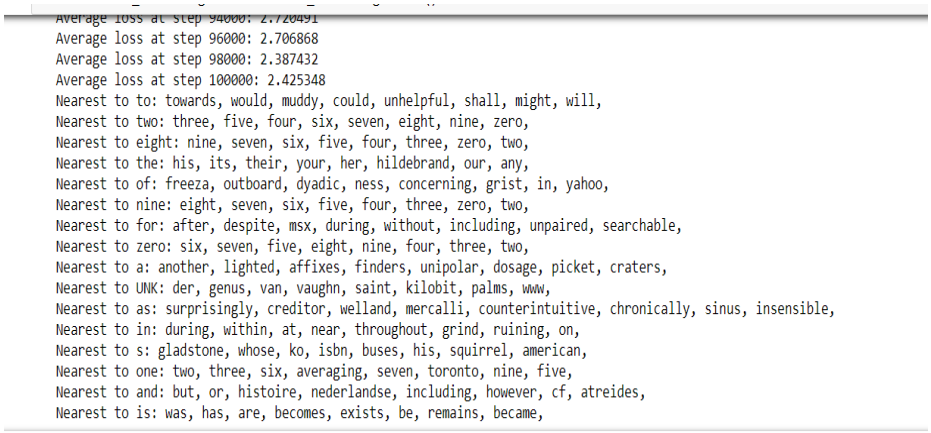

Fig. 3. Efficiency of CBOW model.

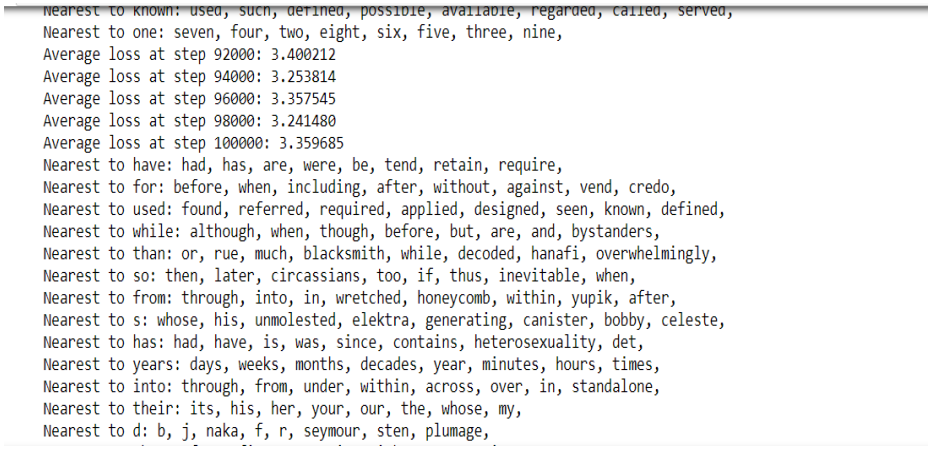

Fig. 4. Efficiency of Skip Gram model.

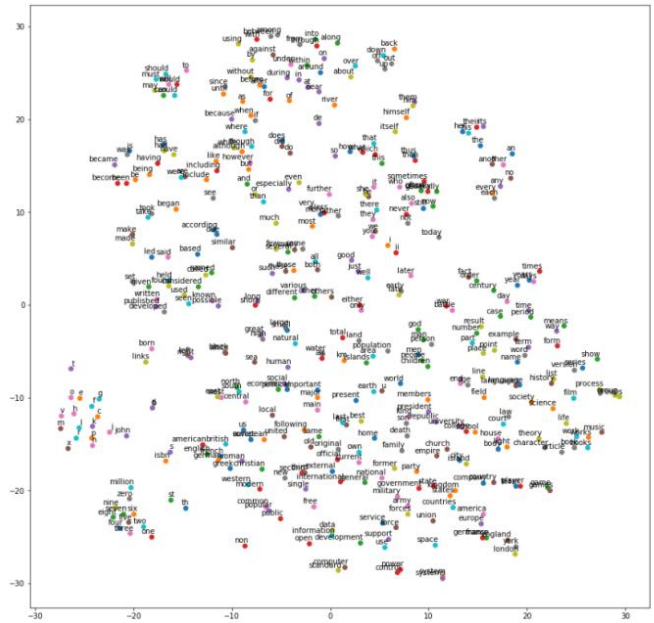

Fig. 5. Graph of words created using CBOW model. We can see here that the words ‘America’, ‘Europe’, ‘London’, ‘countries’, ‘states’, ‘kingdom’ and ‘union’ are mapped close to each other.

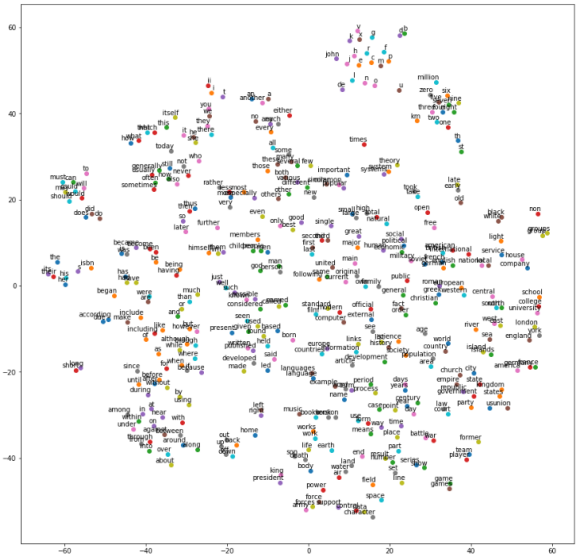

Fig. 6. Graph of words created using Skip Gram model. Here, the words ‘empire’, ‘states’, ‘kingdom’, ‘republic’, ‘government’, ‘church’, ‘party’ and ‘city’ are mapped close while the word ‘America’ is relatively far.

5 Conclusion & Future Work

It has been determined that Continuous Bag of Words has greater efficiency when working with larger datasets while Skip Gram model is preferred for smaller datasets. We have worked with a data size of over 1 million words, which is the reason that the loss computed in Continuous Bag of Words is less as compared to Skip Gram model. Thus, making the CBOW model a more effective model for processing words to compete with the increasing amount of text data being generated every second. Based on analysis, it is also found that both these models capture the syntactic and semantic features to some particular extent. A future development in this area of research is to develop a embedding model which can capture word's semantics and also disambiguate word's sense according to its usage in the sentence provided.

References

1. Ye, Z. & Zhao, H. Syntactic word embedding based on dependency syntax and polysemous analysis, Frontiers Inf Technol Electronic Eng (2018) 19: 524. https://doi.org/10.1631/FITEE.1601846
2. Zhu W, Jin X, Ni J, Wei B, Lu Z (2018) Improve word embedding using both writing and pronunciation. PLoS ONE 13(12): e0208785. https://doi.org/10.1371/journal.pone.0208785
3. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135--146.
4. Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S and Dean, Jeff. "Distributed Representations of Words and Phrases and their Compositionality." In *NIPS* , 3111--3119. : Curran Associates, Inc., 2013.
5. Alvarez, Jon Ezeiza. "A review of word embedding and document similarity algorithms applied to academic text." (2017).
6. Mikolov, Tomas, Chen, Kai, Corrado, Greg and Dean, Jeffrey. "Efficient Estimation of Word Representations in Vector Space." *CoRR* abs/1301.3781 (2013)
7. http://blog.aylien.com/overview-word-embeddings-history-word2vec-cbow-glove/
8. https://iksinc.online/tag/continuous-bag-of-words-cbow/
9. http://mattmahoney.net/dc/