IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

Text Classification: A Comparative Analysis of Word Embedding Algorithms

R. Janani[1], S. Vijayarani[2]

[1,2]Dept. of Computer Science, Bharathiar University, Coimbatore, India

Corresponding Author: janani.sengodi@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i4.818822 | Available online at: www.ijcseonline.org

Accepted: 21/Apr/2019, Published: 30/Apr/2019

Abstract **-** Text classification is the task of allocating the documents into one or more number of predefined categories. In general, this technique is used in the field of information retrieval, text summarization and, text extraction. To perform the classification task, transformation of text into feature vectors is the important stage. The main advantage of this transformation is to discover the most significant words from the document. This process is also known as word embedding, which is used to represent the meaning of words into vector format. The word embedding's are employed in a high dimensional space where the embeddings of similar or related words are adjacent to each other. This main aim of this research work is to classify the text documents based on their contents. In order to achieve this task, in this research work the different word embedding algorithms are used to represent documents. The performance measures are Precision, recall, f-measure and accuracy.

Keywords: Text Classification, Document Representation, Word Embedding, Word2Vec, GloVe, WordRank

I. INTRODUCTION

The process of document classification is to allocate the documents into their predefined category based on their content. Let the assortment of documents $D = d_1, d_2, ..., d_n$ and therefore the predefined classes $C = c_1, c_2, ..., c_n$. Then the classification which assigns the documents d_n into one category $c_n$or more. If the documents are assigned to one category which is called single label classification and the documents are assigned to more than one category is called multi label classification. At this moment, the volume of information over the internet is growing in an exponential way [1]. In order to define the proper category for an unstructured document, the classifier is used to classify the text documents automatically.

Word Embedding is used to represent the meaning of words into vector format. The word embedding's are employed in a high dimensional space where the embeddings of similar or related words are adjacent to each other [2]. Word or sense embeddings can be trained on knowledge graphs, but the most common algorithms learn these vectorial representations just by look over the big corpora. These algorithms rely on a particular supposition that is, words that appear in related contexts have similar meanings. This task is always to factorize a word-word matrix which comprises co-occurrence counts, Point-wise Mutual Information (PMI) or similar metrics. The factor matrices are generally called U and V, which define two distinct embedding spaces. U is a matrix that contains the final word embeddings and V is a temporal set of embeddings which contains the representations used for context words [3].

The rest of the paper is organized as follows: related works on various word embedding algorithms are discussed in section II. In section III, the methodology of this research work is illustrated. The results and discussion on various word embedding algorithms are given in section IV and the conclusion of this research is specified in section V.

II. RELATED WORKS

Word embeddings that offer continuous low-dimensional vector representations of words have been widely studied by NLP communities [4,5]. The last few years have seen the development of word embedding methods purely based on the co-occurrence information from the particular corpus [6,7]. Some studies also pay attention to the semantic knowledge stored in the knowledge bases [8]. For example, refine word representations using relational information from semantic lexicons [8], In [5] represent semantic knowledge as a number of ordinal similarity inequalities of related word pairs to learn semantic word embeddings.

Nowadays, the recent research is associated with directly applying word embeddings into real-world applications. In [9] demonstrated that the globally trained word embedding

underperform corpus and query-specific embeddings for retrieval tasks. They proposed locally training word embeddings in a query-specific manner for the query expansion task. In [10] indicated that the underlying assumption in typical word embedding methods is not equal to the need of IR tasks, and they proposed relevance-based models to learn word illustrations based on query document which is related information, which is the key objective of information retrieval system.

For the sentiment analysis task, [4] refined word embedding to avoid generating similar vector representations for sentimentally opposite words. For the contradiction detection task, [3] developed contradiction-specific word embedding to recognize contradiction relations between a pair of sentences. These studies show that general trained word embeddings cannot be optimized for a specific task, thus, they are likely to be suboptimal. To meet the needs of real-world applications, rational word embeddings should have the ability to capture both the semantics of words and the task-specific features of words.

III. METHODOLOGY

The main aim of this research is to analyze the performance word embedding algorithm for document representation. In order to achieve this task, this research work uses three important word embedding algorithm such as, Word2Vec, GloVe and WordRank algorithm.

A. Preprocessing

Document preprocessing is an essential process in the task of document classification, clustering, topic identification, etc., The preprocessing techniques are applied to the document data set to retrieve the substantial information from unstructured documents. This method will increase the ability of the document classification system [11]. In this research work, stemming, stop word removal, numbers and punctuation removal techniques and normalization techniques are used to retrieve the substantial knowledge.

B. Word Embedding Methods

Word embeddings are a class of methods where singular words are signified to as real valued vectors in a predefined vector space. Each word in a given document is plotted to one vector and the vector values are found out in a way that takes after a neural network, and afterwards the procedure is frequently endured into the field of deep learning [2].

1) Word2Vec

Word2vec is an efficient analytical model is to transform the raw text into word embeddings from raw text. This model is based on the assumption which words with similar semantics present in the same context. This can be modelled by placing a word in a high dimensional vector space and then moving words closer based on their probabilities to appear in the same context. Two important methods are used to calculate these vectors such as, Continuous Bag-of-Words model (CBOW) and the Skip-Gram model [2]. The main advantage of this model is to handle huge volume of documents, it will give the optimal results with word vectors.

The CBoW method is established on the principle of expecting a middle word in a specific context. Here, the context refers n-history and n-future words from the given document, where n is based on the size of the window. The CBoW structure is based on a neural network model. The main objective of CBoW method is to maximize the log probabilities. Though, to feed the network with words, the dictionary has been created with word vectors. This contains a million of words and range of the projection layer between 50 and 1000 nodes [12].

Let the document corpus with word $w_1, w_2, w_3 \dots \dots, w_n$. The window is c and the target value is denoted as t. The objective function is as follows,

$$\frac{1}{T}\sum_{t=1}^{T} \log p\left(w_t \mid \sum_{-c \le j \le c} w_{t+j}\right) \quad (1)$$

The skipgram model is related to the CBoW model but as an alternative of predicting the center word, skipgram predicts the context given the center word. This allows the skipgram model to generate a lot more training data which makes it more suitable for small datasets. The objective function of this method as follows,

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-c \le j \le c} \log p(w_{t+j} \mid w_t) \quad (2)$$

Let E is the number of epochs and S is the size of the corpus. M denoted as the model for transformation. The computational complexity of this algorithm is defined as,

$$= E \; x \; S \; x \; M \quad (3)$$

2) GloVe

The Global Vectors for Word Representation, or GloVe, calculation is an augmentation to the word2vec strategy for efficiently learning word vectors, created by Pennington, et al. at Stanford University. Conventional vector space models expose of words were produced utilizing matrix factorization strategies. GloVe is an approach to extracts both the novel measurements of matrix factorization procedures like LSA with the local context-based learning in word2vec. GloVe constructs an express word-context or word co-occurrence matrix utilizing statistics over the whole text corpus [13]. The outcome is a learning model is the better embeddings in terms of words.

Let i, j is the words and k is the set of probe words. F is the function which is applied to the word vectors. This can be defined as,

$$G = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T w_k + b_i + b_k - \log\left(1 + x_{ik}\right)\right)^2 \quad (4)$$

The upper bound of computational complexity of this algorithm is defined as,

$$O(|V|^2) \quad (5)$$

3) WordRank

WordRank is most commonly used word embedding algorithm, which uses a context window to scan over the document collection and optimize its representation of words. On the other hand, this method optimizes the words in different and novel strategy [14]. This method will train the embeddings such that for each target word, all its context words are ranked by relevance. It is designed to be optimal for retrieving the most similar words to any target word. It also optimizes for precise distinction between the highest ranked similar words. The other word embedding algorithms performs a matrix factorization on the transformed matrix which relates the words to each other [15]. This method retains the window based framework, but it optimizes for the various similarity measures. As an alternative of resembling a pairwise measure between target and context words, it approximates a ranking of contexts per target word [16]. By using this method, all context words are ranked by its relevance.

WordRank estimates the matrix factors U and V. U contains all the embeddings for targets denoted u and V contains all the embeddings for contexts v.

$$J(U,V) = \sum_{w=W} \sum_{c \in w} r_{w,c} . \rho \left(\frac{rank(w,c) + \beta}{\alpha} \right) \quad (6)$$

Where $r_{w,c}$ is the association measure among the word pair (w,c) *and* W is the weight measure for the embedding algorithm. The α and β is the hyper parameters which is used to balance the accuracy.

IV. RESULTS AND DISCUSSION

All the experiments are carried out on a 2.00 GHz Intel CPU with 1 GB of memory and running on windows 10. We implement the algorithm to attain the accurate categories of documents and verified the success of text classification.

A. Datasets

To analyze the performance of this word embedding algorithms, three datasets are used for experimentation. They are, Reuters dataset, 20Newsgroup dataset and 5AbstractsGroup. Reuters -21578 was collected from the Reuters Newswire in the year 1987. It contains 21578 documents with five sets of categories. Each category set contains different number of classes from 39 to 267. The 20newsgroup was collected from 20 different types of newsgroups and the document corpus contains 20 categories with approximately 20000 numbers of documents.

The 5AbstractsGroup dataset is academic papers from five different domains collected from the Web of Science namely, business, artificial intelligence, sociology, transport and law. We extracted the abstract and title fields of each paper as a document. The dataset contains 6,256 documents, and each category contains both training and testing documents. The detailed statistics of all the datasets are listed in Table 1.

Table 1: Dataset Summary

Dataset Name	Type	Train Size	Test Size	Number of Classes	Number of Tokens
Reuters	Doc.	31547	30451	138	13,158,169
20Newsgroup	Doc.	11314	7532	20	6,555,230
5Abstracts Group	Doc.	2500	3756	5	1,203,022

B. Performance Measures

In order to perform this classification task, there are four performance measures are used in this research work. They are precision, recall, f-measure, accuracy of the classification and the learning time [1]. TP denotes the true positive, FP denotes the false positive. True negative is TN and false negative is FN.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{F-Measure} = \frac{2TP}{2TP+FP+FN} \quad (9)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

C. Results

The performance word embedding algorithms on Reuters dataset is shown in Table 2. The training and testing ratio of all the datasets are 70% and 30% respectively. From this, we inferred that the skipgram and GloVe algorithm yields the better accuracy when compared to the other existing techniques. This embedding techniques are differing from one document dataset to another. Generally, the Word2Vec algorithm performs well when the size of the corpus is big.

Table 2: Performance Comparison on Reuters Dataset

Algorithm	Precision	Recall	F-measure	Accuracy
Word2Vec+CBoW	0.694	0.699	0.698	0.791
Word2Vec+ Skipgram	0.705	0.712	0.71	0.804
GloVe	0.701	0.715	0.709	0.804
WordRank	0.702	0.718	0.711	0.802

In Table 3, the comparison of performance values on 20Newsgroup dataset is shown. In this dataset, almost all the algorithms are performed equally especially the CBoW algorithms performance is slightly increased than other techniques. Compared to the Reuters, here the accuracy is slightly decreased. This can be based on the documents and its related words.

Table 3: Performance Comparison on 20Newsgroup Dataset

Algorithm	Precision	Recall	F-measure	Accuracy
Word2Vec+CBoW	0.731	0.732	0.736	0.754
Word2Vec+ Skipgram	0.765	0.749	0.744	0.752
GloVe	0.698	0.699	0.697	0.704
WordRank	0.748	0.758	0.759	0.75

The performance comparison of precision, recall, f-measure and accuracy on 5AbstractsGroup dataset is given in Table 4. For this dataset based on the performance, the GloVe algorithm performs well in terms of accuracy. There is a 2% of increment when compared to the skipgram model. Overall, the accuracy is high when compared to the other dataset values.

Table 4: Performance Comparison on 5AbstractsGroup Dataset

Algorithm	Precision	Recall	F-measure	Accuracy
Word2Vec+CBoW	0.795	0.784	0.799	0.824
Word2Vec+ Skipgram	0.814	0.804	0.814	0.859
GloVe	0.845	0.836	0.844	0.872
WordRank	0.842	0.829	0.830	0.841

The overall performance measures of three datasets are given in Figure 1 to Figure 4. From this graph, we inferred that, the 5Abstract Group datasets yields the better performance in terms of precision, recall, f-measure and accuracy.

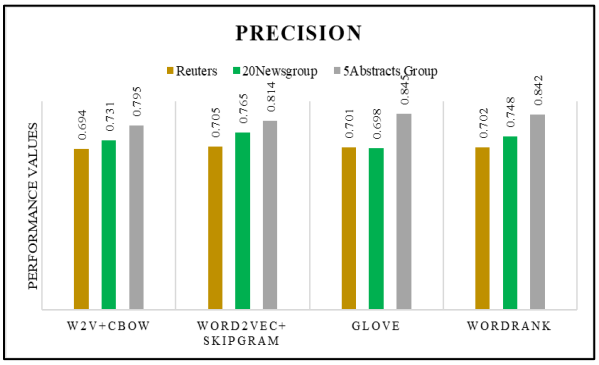

Figure 1: Precision values of three datasets

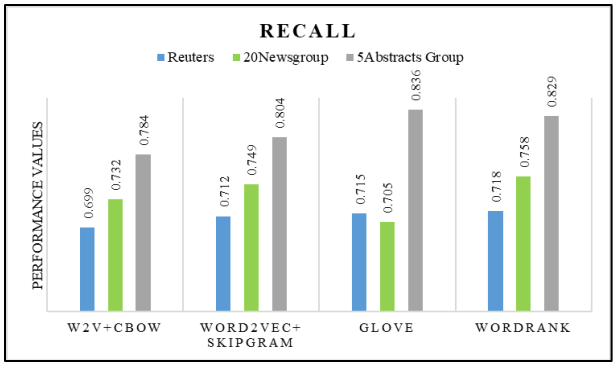

Figure 2: Recall values of three datasets

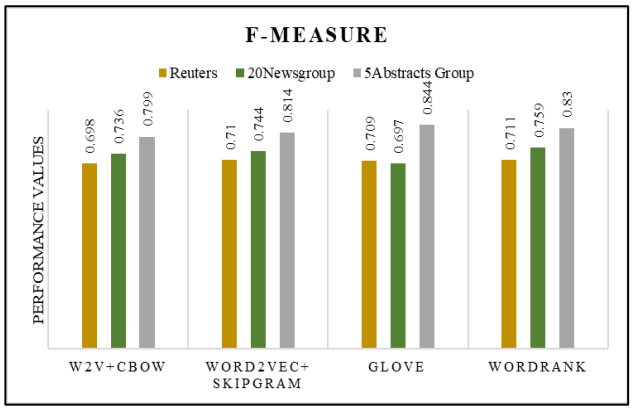

Figure 3: F-Measure values of three datasets

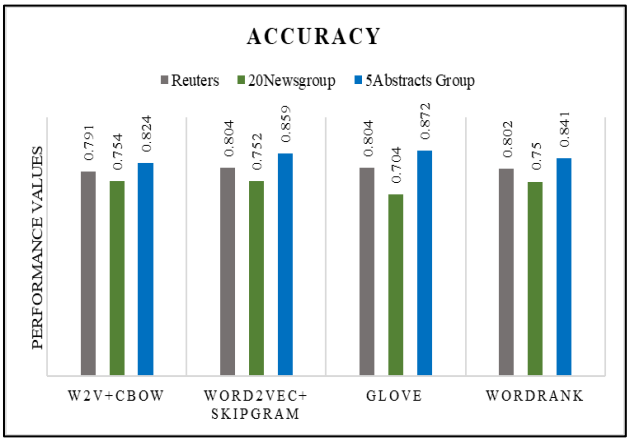

Figure 4: Accuracy values of three datasets

V. CONCLUSION

Text document classification plays vital role in the area of information retrieval, natural language processing and text mining. Word Embedding is used to represent the meaning of words into vector format. The word embedding's are employed in a high dimensional space where the embeddings of similar or related words are adjacent to each other. The main aim of this research work is to analyze the performance of word embeddings algorithm. For this analysis, three most common word embedding algorithms are used for experimentation. The performance measures are precision, recall, f-measure and accuracy. Based on the performance measures, the Word2Vec algorithm gives the better accuracy. In future, the novel techniques to be proposed for word embeddings.

REFERENCES

[1]. Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications, 3(2), 85.

[2]. Jon Ezeiza Alvarez. (2017). A review of word embedding and document similarity algorithms applied to academic text

[3]. Liu, Q., Huang, H., Gao, Y., Wei, X., Tian, Y., & Liu, L. (2018, August). Task-oriented word embedding for text classification. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 2023-2032).
[4]. Yu, L. C., Wang, J., Lai, K. R., & Zhang, X. (2017, September). Refining word embeddings for sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 534-539).
[5]. Li, L., Qin, B., & Liu, T. (2017). Contradiction detection with contradiction-specific word embedding. Algorithms, 10(2), 59.
[6]. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.
[7]. Bollegala, D., Alsuhaibani, M., Maehara, T., & Kawarabayashi, K. I. (2016, March). Joint word representation learning using a corpus and a semantic lexicon. In Thirtieth AAAI Conference on Artificial Intelligence.
[8]. Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. arXiv preprint arXiv:1411.4166.
[9]. Diaz, F., Mitra, B., & Craswell, N. (2016). Query expansion with locally-trained word embeddings. arXiv preprint arXiv:1605.07891.
[10]. Zamani, H., & Croft, W. B. (2017, August). Relevance-based word embedding. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 505-514). ACM.
[11]. Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. Information Processing & Management, 50(1), 104-112.
[12]. Bollegala, D., Yoshida, Y., & Kawarabayashi, K. I. (2018, April). Using k-way Co-occurrences for Learning Word Embeddings. In Thirty-Second AAAI Conference on Artificial Intelligence.
[13]. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
[14]. Dutta, D. (2018). A Review of Different Word Embeddings for Sentiment Classification using Deep Learning. arXiv preprint arXiv:1807.02471.
[15]. Mandelbaum, A., & Shalev, A. (2016). Word embeddings and their use in sentence classification tasks. arXiv preprint arXiv:1610.08229.
[16]. Rosander, O., & Ahlstrand, J. (2018). Email Classification with Machine Learning and Word Embeddings for Improved Customer Support.

