RESEARCH ARTICLE OPEN ACCESS

Performance Analysis of Different Word Embedding Models for Text Classification

Ajose-Ismail B.M.[1], Abimbola O.V.[2], Oloruntoba S.A.[3],

1(Department of Computer Science, The Federal Polytechnic,Ilaro, Ogun State)

2 (Department of Statistics, The Federal Polytechnic,Ilaro, Ogun State)

3 (Department of Computer Science, The Federal Polytechnic,Ilaro, Ogun State)

--************************----------------------------------

Abstract

The task of classifying an unstructured document tothe proper category to which it belongs to is becoming a herculean task because of the steady but exponential growth in the volume of information shared over the internet. 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. From extant literature, the performance of text classification system depends on adequate textual representation of the text document. To perform the classification task, transformation of text into feature vectors is a very important stage. Several textual representation techniques such as bag of words, n-gram and topic models have been proposed by authors to capture the real semantics of web documents but are fraught with several challenges such as semantic mismatch and multiple meanings of words.

Thus, this paper proposes word embedding's to solve the document representation problem in text classification systems. In order to achieve this task, this research work utilizes different word embedding algorithms to represent documents which are also used in conjunction with classification algorithms to determine the most effective embedding model. Results obtained confirms the earlier assumption that Word2Vec performs robustly on very high dimensional text such as web documents, it also captures the real semantics of the web document The performance metrics employed in this research work are Precision, f-measure and accuracy.

.

Keywords — Text Classification, Document Representation, Word Embedding, Word2vec, XgBoost, TF-IDF

--************************----------------------------------

I. INTRODUCTION

Text classification is a construction problem of models which can classify new documents into pre-defined classes [1],[2]. It involves assigning documents into their predefined categories based on their contents. Let the set of Documents, $D = \{d_1, d_2, \dots, d_k\}$ and therefore the predefined classes $C = c_1, c_2, \dots, c_n$ .Classification then involves assigning the documents d_k into one category c_n or more..Classification could occur in either one of two ways viz single label classification and multi-label classification. In single label classification, the documents are assigned to one category, whereas in multi-label classification, documents are assigned to more than one category.

Recently, there has been a steady but exponential growth in the volume of information shared over the internet [3]. To define the proper category for which an unstructured document belongs, the documents needs to be represented in a way that would enable a classifier to classify the text documents automatically.

Word embedding is a feature learning technique in which each word or phrase from the vocabulary is mapped to an N dimension vector of real numbers [4]. The word embeddings are employed in a high dimensional space where the embedding's of similar or related words are adjacent to each other [5]. Learning word embedding's usually relies on the distributional hypothesis – words appearing in similar contexts must have similar meanings, and thus close representations. Finding such representations for words and sentences has been one hot topic over the last few years in Natural Language Processing (NLP) and has led to many improvements in core NLP tasks [2][6].

In this paper, we seek to compare the effect of different word or document representation i.e. word embeddings for text classification tasks.

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

In [8], the authors proposed a novel semantic hierarchy for short texts modelling and classification. The pre-trained words embeddings were used to initialize the lookup table, which introduced extra knowledge and enabled the authors' measure words affinity by computing the Euclidean distance between two vector representations. The additive composition method was utilized to compute multi-scale semantic units for short texts expansion. In the embedding spaces, similar words are grouped together that help learning algorithms to achieve better performance.

In [9], a novel approach for learning task-oriented word embedding, especially for the text classification task was proposed. Instead of learning embedding vectors merely based on context information, the authors incorporated task-specific features into the training process in order to reveal the words functional attributes in the embedding space

[10], developed a method using Word2Vec to reduce the feature size while increasing the classification accuracy. They achieved feature reduction by loosely clustering similar features using graph search techniques. Similarity thresholds above 0.5 was used in their method to pair and cluster the features. Finally, they utilized Multinomial Naïve Bayes classifier, Support Vector Machine, K-Nearest Neighbour and Random Forest classifier to evaluate the effect of their method

In [11], used a Twitter election classification task that aims to detect election-related tweets to investigate the impact of the background dataset used to train the embedding models, as well as the parameters of the word embedding training process, namely the context window size, the dimensionality and the number of negative samples, on the attained classification performance.

[12], attempted to add word-cluster embedding to deep neural network for improving short text classification. Initially, hierarchical agglomerative clustering was used to cluster the word vectors in the semantic space. Then the authors proceeded to calculate the cluster center vector which represents the implicit topic information of words in the cluster. Finally, they expanded word vector with cluster center vector, and implemented classifiers using CNN and LSTM respectively.

III.METHODOLOGY

This section describes the phases involved in analysing the performance of different word embedding models in text classification tasks. In order to achieve this task, this research work uses four important word embedding algorithm namely, Word2Vec, GloVe, Count Vectorizer and TF-IDF algorithm.

A. *Preprocessing*

Data preprocessing is the first and most important thing to do in other to achieve a great accuracy during model building. Document preprocessing is an essential process in the task of document classification, clustering, topic identification, amongst others. For the purpose of this research, we applied different data preprocessing based on the nature of the dataset we gather from different sources. For all the dataset employed in this research, stop words were removed, we normalized all our dataset by transforming word which are wrongly spelt or shortened into the normal form, HTML tags and emoji's are removed, special character were removed, contractions were expanded and all text were converted to lower text.

B. *Word Embedding Models*

Word embeddings are in certainty a class of methods where singular words are represented to as real valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are found out in a way that takes after a neural network, and subsequently the procedure is frequently lumped into the field of deep learning [13]

1) *Word2Vec:*

Word2Vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text. It consists of a shallow, two-layer neural networks which is trained to reconstruct linguistic contexts of words. It takes as its input a large corpus of words and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. It comes in two variants, the Continuous Bag-of-Words (CBOW) model and the Skip-Gram model [5]. CBOW predicts target words (e.g. 'mat') from the surrounding context words ('the cat sits on the'). Skip-gram predicts surrounding context words from the target words (inverse of CBOW).

2) *GloVe:*

The Global Vectors for Word Representation, or GloVe is an augmentation to the word2vec strategy for efficiently learning word vectors, created by Pennington, et al. at Stanford University. GloVe is an approach to extract 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 [Pennington, 2014]

3) *Count Vectorizer:*

The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. An encoded vector is returned with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document. These vectors often contain lots of zeros, we call them sparse. Bag-of-words models encode every word in the vocabulary as one-hot-encoded vector e.g., for the vocabulary of size $|\sum|$, each word is represented by $|\sum|$, dimensional sparse vector with 1 at index corresponding to the word and 0 at every other index. Count Vectorizer is used for the Bag-of-words-model [4]

4) ***TF-IDF:***

[14] proposed Inverse Document Frequency (IDF) as a method to be used in conjunction with term frequency in order to lessen the effect of implicitly common words in the corpus. IDF assigns a higher weight to words with either high or low frequencies term in the document. This combination of TF and IDF is well known as Term Frequency-Inverse document frequency (TF-IDF). The mathematical representation of the weight of a term in a document by TF-IDF is given in Equation (1).

$$W(d, t) = TF(d, t) * \log \frac{N}{df(t)} \quad . . .(1)$$

Here N is the number of documents and d f (t) is the number of documents containing the term t in the corpus. The first term in Equation (1) improves the recall while the second term improves the precision of the word embedding [15]

IV. RESULTS AND DISCUSSION

To overcome the constraint of computational power, all of the experiment were performed using Google Colab GPU and implemented with the classification algorithms to ascertain the accurate classification of documents and verify the success of text classification.

A. Datasets

To analyze the performance of this word embedding algorithms, four datasets were used for experimentation. These datasets were obtained from Zindi and Kaggle repository. The first dataset we used is the IMDB dataset [16] having 50000 movie reviews for natural language processing or text analytics which contains a target class of "Positive or Negative", followed by the Health Status dataset [17] which contains 616 statements and questions expressed by students from multiple universities across Kenya who reported suffering from these different mental health challenges with a target class of "Depression, Alcohol, Suicide and Drugs". The third data is about Stock Market Sentiment [18] which are gathered from different twitter handles on 5791 stock tweets with a target class of "Positive and Negative Sentiment", while the last is a Vaccination data collected through crowdsourcing on 10001 tweets related to vaccinations with a target class of positive, neutral, or negative [19].

B. Performance Measures

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

C. Results

The performance word embedding algorithms on IMDB dataset is shown in Table 1. The training and testing ratio of all the datasets are 80% and 20% respectively

Table 1: Performance Comparison of IMDB Dataset

Algorithm /Embedding Model	Performance Metric	Word2Vec	Glove	Count Vectorizer	TF-IDF
LR	Accuracy	0.86	0.76	0.82	0.84
	Precision	0.86	0.76	0.81	0.84
	F-Measure	0.86	0.76	0.82	0.84
RF	Accuracy	0.82	0.76	0.80	0.80
	Precision	0.82	0.76	0.80	0.80
	F-Measure	0.81	0.76	0.79	0.79
SVC	Accuracy	0.86	0.75	0.79	0.84
	Precision	0.87	0.75	0.78	0.85
	F-Measure	0.86	0.75	0.79	0.84
XGB	Accuracy	0.83	0.76	0.83	0.81
	Precision	0.83	0.76	0.84	0.81
	F-Measure	0.83	0.76	0.83	0.81

Table 1 above shows the model comparison of the performance of each word embedding and machine learning algorithm used on the IMDB dataset. Looking closely at the table, it was observed that Word2Vec appears to be a better word embedding tool which assisted the modelling of IMDB dataset. Based on the result above, it was shown that the use of Support Vector Machine with Word2Vec as an embedding tool works better than the other modelling approach. Support Vector Machine has the highest Classification accuracy of about 86% which is better than that of Random forest, and Xgboost. Logistic Regression has the same classification accuracy but SVC has a higher precision which makes it superior. The graphical representation showing the precision, f-measure and accuracy of the embedding model when used with the classification algorithm is shown in Fig 1.

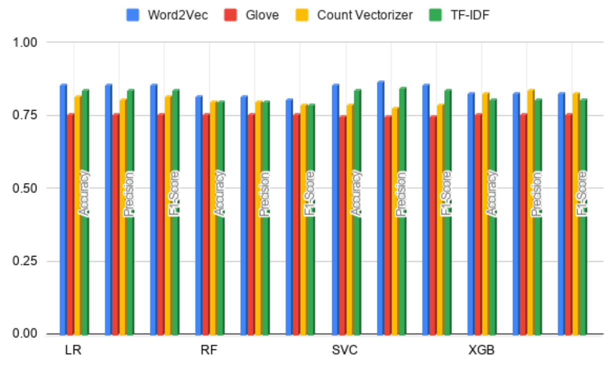

Fig 1: IMDB Model Performance

Table 2: Performance Comparison of Health Status Dataset

Algorithm/ Embedding Model	Performance Metric	Word2Vec	Glove	Count Vectorizer	TF-IDF
LR	Accuracy	0.91	0.74	0.87	0.82
	Precision	0.89	0.73	0.88	0.85
	F-Measure	0.90	0.73	0.86	0.80
RF	Accuracy	0.85	0.74	0.83	0.83
	Precision	0.82	0.72	0.83	0.84
	F-Measure	0.84	0.7	0.81	0.82
SVC	Accuracy	0.8	0.68	0.79	0.81
	Precision	0.83	0.56	0.83	0.84
	F-Measure	0.82	0.6	0.77	0.79
XGB	Accuracy	0.85	0.76	0.85	0.83
	Precision	0.84	0.74	0.85	0.82
	F-Measure	0.8	0.74	0.84	0.82

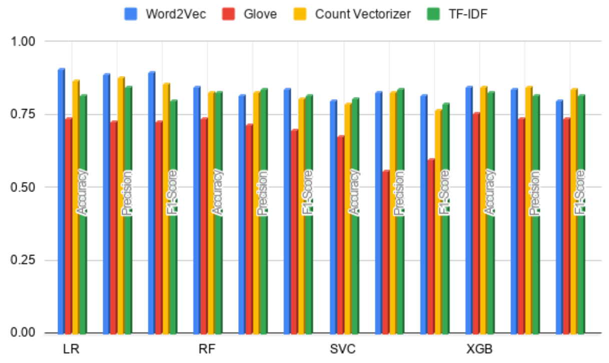

Fig 2: Health Status Model Performance

Table 3: Performance Comparison of Stock Dataset

Algorithm/ Embedding Model	Performance Metric	Word2Vec	Glove	Count Vectorizer	TF-IDF
LR	Accuracy	0.88	0.76	0.79	0.84
	Precision	0.88	0.76	0.79	0.84
	F-Measure	0.88	0.76	0.79	0.84
RF	Accuracy	0.84	0.76	0.78	0.8
	Precision	0.84	0.76	0.78	0.8
	F-Measure	0.83	0.76	0.77	0.79
SVC	Accuracy	0.88	0.75	0.77	0.84
	Precision	0.89	0.75	0.78	0.85
	F-Measure	0.88	0.75	0.75	0.84
XGB	Accuracy	0.85	0.76	0.75	0.81
	Precision	0.85	0.76	0.75	0.81
	F-Measure	0.85	0.76	0.73	0.81

Applying the same method on the Stock Dataset we achieve a highest accuracy of 88% using Word2vec, 76% using glove, 79% using Count Vectorizer and 84% using TF-IDF. On the other hand, we noticed that generally applying the four Machine learning models, Word2Vec is a better embedding model in classifying the sentiment of the stock dataset while its graphical representation is shown in Figure 3 below.

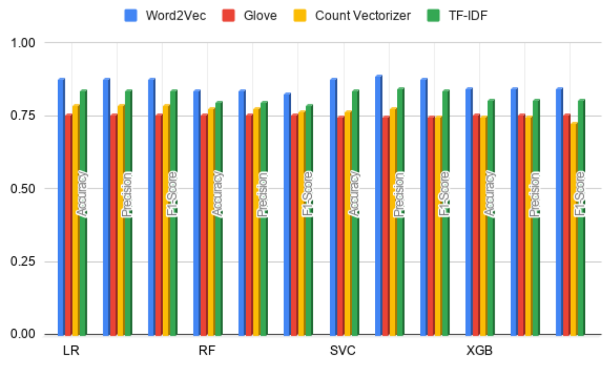

Fig 3: Stock Dataset Model Performance

Table 4: Performance Comparison of Vaccination Dataset

Algorithm/ Embedding Model	Performance Metric	Word2Vec	Glove	Count Vectorizer	TF-IDF
LR	Accuracy	0.83	0.64	0.73	0.73
	Precision	0.83	0.62	0.72	0.73
	F-Measure	0.81	0.61	0.73	0.71
RF	Accuracy	0.83	0.67	0.74	0.73
	Precision	0.83	0.65	0.73	0.73
	F-Measure	0.80	0.63	0.72	0.70
SVC	Accuracy	0.84	0.65	0.74	0.74
	Precision	0.84	0.59	0.75	0.74
	F-Measure	0.81	0.62	0.72	0.71
XGB	Accuracy	0.82	0.66	0.73	0.72
	Precision	0.82	0.70	0.73	0.72
	F-Measure	0.79	0.63	0.71	0.69

By comparing the performance of the vaccination dataset it is obvious from Fig 4 below that the performance of Word2Vec embedding model on the three machine learning models is much better than others. However, it was observed that the performance of Count Vectorizer and TFIDF is almost of the same range while the performance of glove is not quite up to the rest.

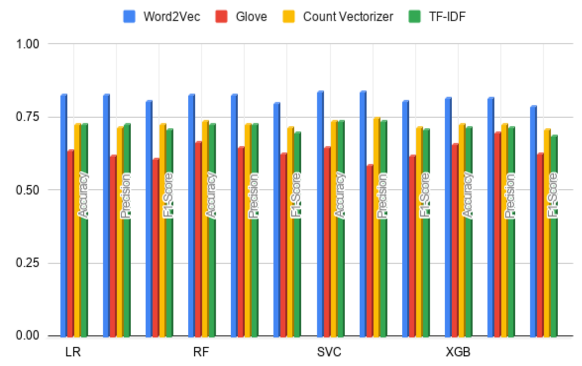

Fig 4: Vaccination Dataset Model Performance

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. We have been able to analyze the performance of different word embeddings algorithm for text classification. For this analysis, four different word embedding algorithms are used for experimentation. Looking at the performance generally based on our experiment, it is very obvious that Word2vec is a better embedding tools in text classification as it perform better than Glove, Count Vectoriser and TF-IDF. Also we observed that the performance was based on different Machine learning models. Therefore, we could not conclude that a particular model could word better in Text classification. However, we strongly recommend for further researcher's that the comparison of more dataset and word embedding tools will help draw more patterns in understanding the better embedding tools that could be better in text classification. However the use of more embedding tools, the use of transformer embedding tools and the use of deep learning models should not be left out.

REFERENCES

[1]. Liu, B. (2007). Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media. Secaucus, NJ, USA: Springer-Verlag New York, Inc

[2]. Pennington, J., Socher, R., & Manning, C. D. (2014,). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

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

[4] Kowsari, K., JafariMeimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

[5] Alvarez, J. E., &Bast, H. (2017). A review of word embedding and document similarity algorithms applied to academic text. Bachelor thesis.

[6] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[7] Manning, C. D., Raghavan, P., &Schütze, H. (2008). Introduction retrieval. New York, NY, USA: Cambridge University Press.

[8] Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. L., &Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing, 174, 806-814.

[9] Liu, Q., Huang, H. Y., Gao, Y., Wei, X., Tian, Y., & Liu, L. (2018). Task-oriented word embedding for text classification. In Proceedings of the 27th international conference on computational linguistics (pp. 2023-2032).

[10] Ge, L., &Moh, T. S. (2017). Improving text classification with word embedding. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 1796-1805). IEEE.

[11] Yang, X., Macdonald, C., &Ounis, I. (2018). Using word embeddings in twitter election classification. Information Retrieval Journal, 21(2-3), 183-207.

[12] Shen, Y., Zhang, Q., Zhang, J., Huang, J., Lu, Y., & Lei, K. (2018). Improving medical short text classification with semantic expansion using word-cluster embedding. In International Conference on Information Science and Applications (pp. 401-411). Springer, Singapore.

[13] Dutta, D. (2018). A Review of Different Word Embeddings for Sentiment Classification using Deep Learning. arXiv preprint arXiv:1807.02471.

[14] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of documentation.

[15] Tokunaga, T., & Makoto, I. (1994). Text categorization based on weighted inverse document frequency. In Special Interest Groups and Information Process Society of Japan (SIG-IPSJ.

[16] "IMDB dataset of 50K movie reviews." *Kaggle: Your Machine Learning and Data Science Community*, www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

[17] "Basic Needs Basic Rights Kenya - Tech4MentalHealth." Zindi, www.zindi.africa/competitions/basic-needs-basic-rights-kenya-tech4mentalhealth/data.

[18] "Stock-Market Sentiment Dataset." *Kaggle: Your Machine Learning and Data Science Community*, www.kaggle.com/yash612/stockmarket-sentiment-dataset.

[19] "#ZindiWeekendz Learning: To Vaccinate or Not to Vaccinate: It's Not a Question." *Zindi*, www.zindi.africa/competitions/zindiweekendz-learning-to-vaccinate-or-not-to-vaccinate/data.