Bengali Word Embeddings and It's Application in Solving Document Classification Problem

Adnan Ahmad
Researcher, Search Engine Pipilika
Department of Computer Science and Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh.
adnan.ahmad@student.sust.edu

Mohammad Ruhul Amin
PhD student, Computer Science Department
Stony Brook University
NY 11790, USA
moamin@cs.stonybrook.edu

***Abstract*—In this paper, we present Bengali word embeddings and it's application in the classification of news documents. Word embeddings are multi-dimensional vectors that can be created by exploiting the linguistic context of the words in large corpus. To generate the embeddings, we collected Bengali news document of last five years from the major daily newspapers. Word embeddings are generated using the Neural Network based language processing model Word2vec. We use the vector representations of the Bengali words to cluster them using K-means algorithm. We show that those clusters can be used directly to perform various natural language processing task by solving the problem of Bengali news document classification. We use the Support Vector Machine (SVM) for the classification task and achieve ~91% F1-score. The accuracy of our method demonstrates that our word embeddings could capture the semantics of word from the respective context correctly.**

Keywords— Bengali, Word Embedding, Word2vec, Document Classification, Word Cluster

I. Introduction

In the recent years, word embeddings or the vector representation of the words have been proved to achieve significant performance in the language modeling and in the natural language processing (NLP) tasks [1]. The word embedding of a word represent the word in a multi-dimensional space in which the semantically similar words are placed closer to each other and non-related words are placed far from one another [2][3][4]. Thus, these distributed vector representations can be used to learn the abstract relationship among the words by using unsupervised clustering methods. The features of those clusters can be used very effectively to solve various NLP tasks like document classification, sentiment analysis, parts-of-speech tagging, named entity recognition and machine translation etc [1][4].

Bengali is a highly inflected as well as morphologically rich language [5]. A slight modification in a word can change it's form to express a completely different meaning from the original one in terms of tense, mood, person, number and gender to name a few [5]. So, clustering words that shares similar concepts in Bengali is a very challenging task. Very few attempts are taken to cluster Bengali words. Those attempts are mainly based on the N-gram language model [6], where clusters are generated by considering the words with their frequency in a context up to trigram. The N-gram model only consider the consecutive words and their relative frequencies in a N-gram window. The probability of a word in the context is calculated only from the context of previous words. This probability cannot be used to represent the distance or similarity among all the words in a language. Thus, N-gram model cannot be used directly for clustering the semantically similar words together, let alone solving the other NLP problems in Bengali.

In this paper, we present the application of Begnali word embeddings to solve document classification problem in Bengali. We create vector representation of Bengali words using Word2vec model [2]. We use t-SNE, an efficient dimension reduction technique to map those multi-dimensional vectors into two-dimensional space [7]. We then apply K-means clustering to find the clusters of word embeddings, those are found in close proximity in the multi-dimensional space [8]. Finally, We use the cluster information of Bengali word embeddings as features to solve the problem of Bengali news document classification by using the machine learning algorithm, support vector machine (SVM) [9]. Our model achieve the accuracy of ~91% which justifies that the model can be be used successfully in solving many other NLP problems in Bengali. Specifically, our contributions include:

Largest Collection of Bengali word embeddings: We are going to release the largest collection of Bengali word embeddings. To our knowledge, the only available word embeddings for Bengali were published under the Polyglot project from the Data Science Lab at State University of Stony Brook [4]. Polyglot used the Bengali contents in Wikipedia and created word embeddings for ~55,000 words. For our work, we collected news contents of last five years from 13 major newspapers and analyzed ~52,000,000 of lines to release word embeddings for ~210,000 Bengali words.

Document Classification without Preprocessing: We show that clustering information of Bengali words embeddings can be used as a feature to solve Bengali document

classification problem. Previously, it was considered that accuracy of stemming and key word identification need to be improved for preprocessing the document for better document classification. But, in this paper our method shows that we can use the word embeddings directly for news document classification; hence, we show that document classification can be done independently from the other preprocessing steps.

II. Background Study

A. Bengali Word Embeddings

Word-vectors or so-called distributed representations have a long history by now, starting perhaps from work of S. Bengio et al [10] where he obtained word-vectors as by-product of training neural-net language model. A lot of related researches demonstrated that these vectors do capture semantic relationship between words [11]. Word2vec is a popular word embedding model which is created by using a two layer Neural Network (NN) and skip-gram technique and successfully used for many NLP tasks [1][2]. There are few other popular word embedding models, namely, Polyglot, Glove and Gensim [3][4][12]. To the best of our knowledge, only Polyglot published the word embeddings for ~55,000 Bengali words by from the Bengali wikipedia. Abhishek et al. created a neural lemmatizer using Bengali word embeddings generated by Word2vec model [13] using a relatively small dataset.

B. Bengali Word Clustering

A pioneer work on word clustering is proposed by Brown et al, where they used n-gram language model [14]. Brown clusters have been used successfully in a variety of NLP applications [15]. Another attempt using n-gram model is reported by Korkmaz et al; they used a similarity function and a greedy algorithm to put the words into clusters [16]. Ding et al presented Naive Bayes method for English in classifying words using surrounding context words as features [17]. Further, many other approaches have been reported in literature for other languages like Russian, Arabic, Chinese and Japanese. As mentioned earlier, very few works has been done on Bengali word clustering so far. Tanmoy et al proposed semantic clustering of words using synset to identify Bengali multi-word expressions [18]. Sabir et al proposed an unsupervised machine learning technique to develop Bengali word clusters based on their semantic and contextual similarity using N-gram language model [6].

C. Bengali Document classification

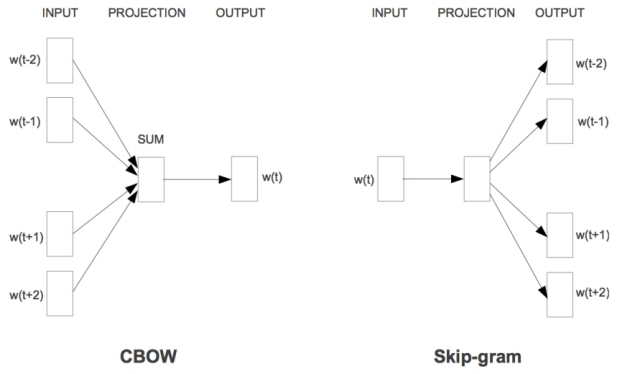
For text classification in other languages, i.e. English, Chinese, Hindi, Arabic and European languages, various number of supervised learning techniques has been used, such as Association Rules [19], Neural Network [20], K-Nearest Neighbour [21], Decision Tree [22], Naïve Bays [23], Support Vector Machine [24], and N-grams [25] etc. Previous works on document classification for Bengali are mainly based on N-gram [26], Naïve Bays [27], Stochastic Gradient Descent based classifier [28]. The features of word clusters have been used to perform various NLP tasks for a long time. We came across a work of Y. Yuan et al, who used word clusters created from Word2vec to perform document clustering in Chinese language by applying Support Vector Machine (SVM) [29].

III. Methodology

A. Neural Network and Word2vec

Words occurring in the same or similar contexts tend to convey similar meaning. There are many approaches to computing semantic similarity between words based on their distribution in a corpus. Word2vec models are shallow, two-layer neural networks which is trained in the unsupervised fashion to reconstruct linguistic context of words. Word2vec takes a large corpus of text as input for training and produces a set of vectors called embeddings, typically of several hundred dimensions, with each unique word in the corpus. Given enough data, usage and contexts, Word2vec can make highly accurate guesses about a word's meaning based on past appearances. Word2vec produces word embeddings in one of two ways: either using context to predict a target word, a method known as continuous bag of words, or CBOW; or using a word to predict a target context, which is called Skip-gram (Figure 1).

Fig. 1. Two ways to compute Word2vec model: 1. Continuous Bag of Words (CBOW) and 2. Skip-gram.

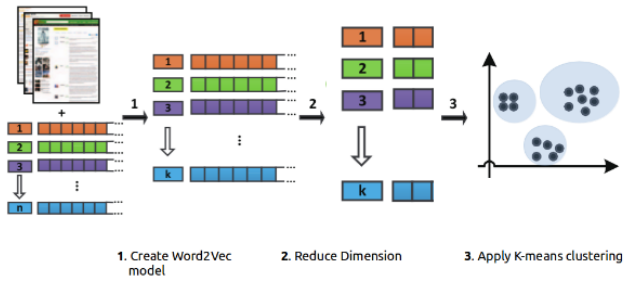


To generate Bengali word embeddings, we use the skip-gram method because Skip-gram works well with small amount of the training data and can represent well even rare words or phrases [2]. In our work, we create a Bengali Word2vec model which contains online newspaper articles from 13 different newspaper of year 2010-2015, a collection of 2185701 documents. To our best knowledge, this is the largest Word2vec model for Bengali language. We create two separate Word2vec models of dimension size 100 and 200, using default parameters. Also, we learned the vector for unknown word (UNK). Later, we use those word vectors to create word clusters for Bengali and use those clusters in document classification task.

B. *Dimensionality Reduction and Clustering*

As Word2vec model represents words as vector, we can directly apply K-means clustering algorithm on top of it. But applying K-means and performing all the calculations in high dimensional feature space is time-consuming. So, before applying K-means, we use dimension reduction technique called t-distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimension size of the vectors into two [7]. It is a nonlinear dimensionality reduction technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scatter plot. This benefits the process in two ways: first, it takes less time to compute the clusters using K-means; second, clusters can be plotted and visualized into a 2D plane. K-means clustering is a method of vector quantization, that is popular for cluster analysis in data mining. The whole process is represented in the Figure 2.

Fig. 2. Clustering the word embeddings: 1. Create word embeddings for each word in a corpus, 2. Reduce the vector dimension, 3. Create clusters of vector s representing words.



C. *Support Vector Machine (SVM) as Document classifier*

For Document classification task, we use Support Vector Machine (SVM) classification algorithm. SVM is a popular supervised learning algorithm for classification task and many researcher attempted to perform document classification task using it [30]. Given a set of training documents, each document marked with a particular category, an SVM training algorithm builds a model that assigns new examples into one of the predefined categories.

Fig. 3. The process of training SVM classifier.

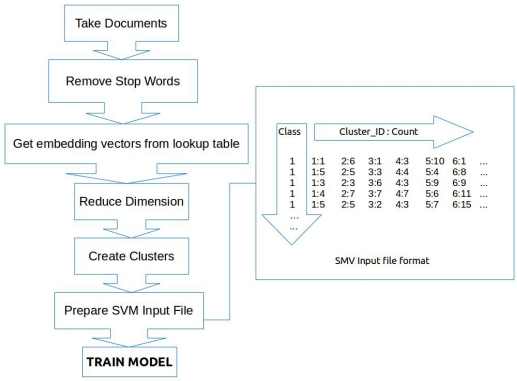


Figure 3 shows the process of training SVM classifier using word clusters as feature. Each row represents a document. First column represents the category id. Rest of the columns represents cluster id's and how many words of a particular document belong to a certain cluster. The model was trained using the default parameters.

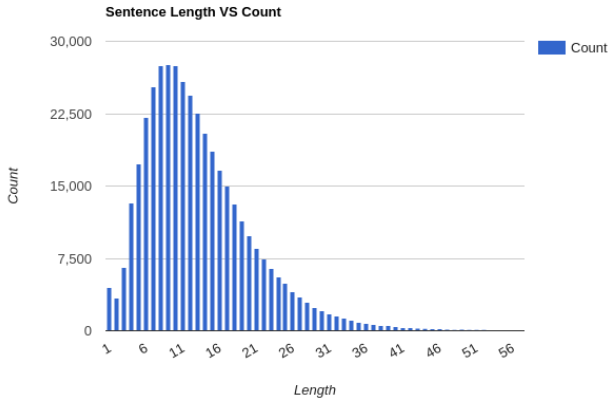
IV. Experiments

Our experiment is completed in three steps. Firstly, we strip off the html tags for all the news crawled from Bengali online newspapers. We use those data to train the Word2vec model and generate embeddings. Secondly, we apply dimension reduction technique and K-means clustering on subset of words to create clusters. Finally, using word clusters as features, we perform document classification task to evaluate the word embeddings and clusters.

A. *Data for Word2vec*

In general, Word2vec model takes a huge amount of data (typically about 100 billion words) as training text to create accurate models; but there are not much Bengali data available online. We collected online newspaper articles from 13 different Bengali newspapers of year 2010 to 2015. Total number of article is 218,5701, totalling 51,920,010 sentences. Most of the sentences contain 5 to 25 words. For the Word2Vec model, we only took words which occurred at least 5 times in the documents, totaling 210,535 words. Figure 3 shows the frequencies of sentences with various sentence lengths.

Fig. 4. Sentence length VS count



B. *Data for Document Classification*

To perform document clustering, we collected ~20,000 Bengali online newspaper documents, each labeled into its particular class. We use 7 general classes like Sports, Entertainment, Politics etc. Overview of the data is given in Table I. We separate 70% document of each class for training and 30% document of each class for testing.

TABLE I. TOTAL NUMBER OF DOCUMENTS FOR CLUSTERING

Class	Class name	Number of documents
0	Sports	2232
1	Entertainment	2655
2	Accident and Crime	4136
3	International	2250
4	Science & Tech.	2906
5	Politics	2808
6	Economics	2718

C. *Clustering Word Embeddings for Document Classification*

Using the training data mentioned above, we train our Word2vec model. We use deeplearning4j[1], a java implementation of Word2vec model, using default experiment setup with the context window size 5 and min word frequency 5. We created two different models with vector size of 100 and 200 for our experiment. Vocabulary size of the final model is 210,535. As words are represented as vectors in a Word2vec model, that makes each word independent of their contexts. We can take any two words and calculate distance or similarity between them. That means, we can use the whole corpus or any subset of words from the corpus to cluster the words by directly applying K-means clustering algorithm. But before applying K-means to those word vectors, first we reduce the dimension size of the vectors to two, using t-SNE dimension reduction technique and then apply K-means. We use R implementation of both the t-SNE[2] and K-means[3]. We create several experiments for the different embeddings size and number of clusters, using K={100, 200, 300, 400, 500, 600}. For each document, we use the number of words in a particular cluster as features to perform SVM classification. We use Scikit-learn libSVM[4], a popular python implementation of SVM to perform multi class classification. We discuss the outcome of our experiments in the result sections.

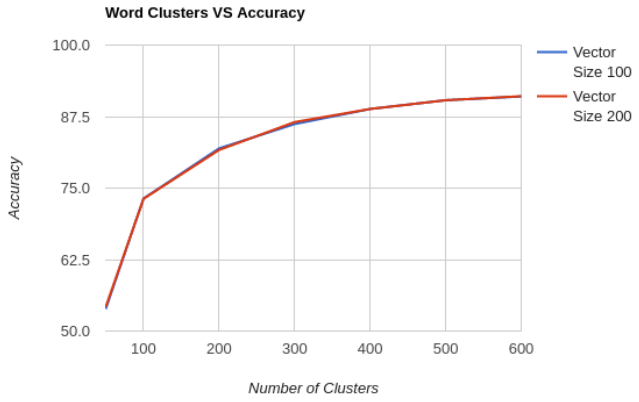
V. RESULTS

In order to evaluate the word clusters, three methods can be used, i.e. measuring the internal coherence of clusters, embedding the clusters in an application, or evaluating against a manually generated answer key [31]. The first method is generally used by the clustering algorithms themselves. The second method is especially relevant for applications that can deal with noisy clusters and avoids the need to generate answer keys specific to the word clustering task. The third method requires a gold standard such as WordNet[32] or some other ontological resource. English and a number of other languages have resources such as WordNet [33][34]. Unfortunately, there exists no WordNet for Bengali words. In order to evaluate the clusters, we perform a NLP task, which is Bengali Document Classification, by using the information of word clusters as features and measured the accuracy of that task.

In Figure 4, we show the graph of performance measure using Word Clusters VS Accuracy, for both the embeddings size of 100 and 200, and K={100, 200, 300, 400, 500, 600}. We achieve our best result, ~91.02% F1-score using K= 600 for both the embedding sizes of 100 and 200. As we reduce the embedding dimension into two before clustering, we observe no significant effect of the number of actual embedding dimension on clustering as well as classification task. But, we must also mention that for the embedding size less than 100, word embeddings did not result in meaningful clusters; meaning contextually unrelated words showed up in the same cluster which resulted in poor classification performance. We also observed such problem while using Polyglot for creating Bengali embeddings. Polyglot uses only 64 dimensions for the embeddings which failed to capture the contexts in Bengali language.

In Figure 5, we show that word clusters become more meaningful and accurate when the number of K in K-means is large. When we cluster the words with relatively small value of K (K=50), the words of the clusters become generalized for which semantic and contextual similarity is hard to relate. But when the value of K is large (K =600), we see more accurate and meaningful clusters are constructed.

Fig. 5. Cluster size VS Classification accuracy



In our classification, the news can be classified into seven classes. For each of those classes, we measure the precision, recall and f1-score and show in Table II for the experimental setup of D=100 and K= 600. Our test data contains 4713 documents, which is 30% of the total dataset and separate from the training set. Here, the result shows average precision of 91%, recall of 90% and F1-score of 91%.

1 http://deeplearning4j.org/word2vec

2 https://lvdmaaten.github.io/tsne/

3 https://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html

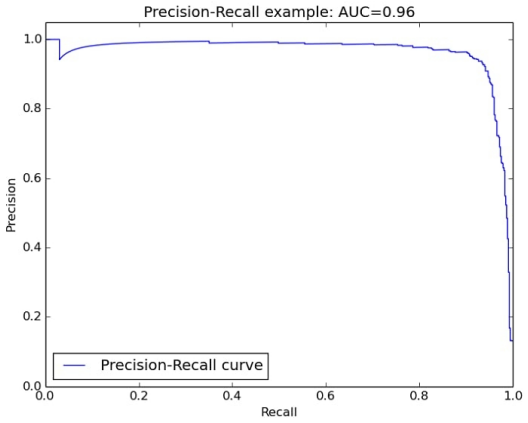
4 http://scikit-learn.org/stable/modules/svm.html

TABLE II. CLASSIFICATION REPORT (D = 100, K = 600)

Class	Precision	Recall	F1-score	Test Document
Sports	0.98	0.94	0.96	528
Entertainment	0.93	0.93	0.93	627
Accident and Crime	0.92	0.91	0.91	996
International	0.90	0.89	0.89	566
Science & Tech.	0.91	0.86	0.88	677
Politics	0.93	0.87	0.90	653
Economics	0.77	0.92	0.84	654
avg / total	0.91	0.90	0.91	4713

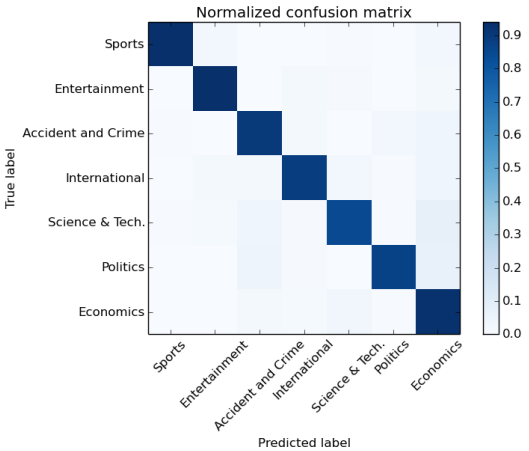
We evaluate our approach with the TREC evaluation technique to produce Precision-Recall graph [35]. In Figure 6, we show the Precision Recall graph for the experiment mentioned above.

Fig. 6. Overall Precision-Recall curve of Document Classification (D = 100, K = 600)



In Figure 7, we present Confusion Matrix to elucidate the performance of our classification model on test dataset. The figures show the confusion matrix with normalization by class support size.

Fig. 7. Normalized Confusion Matrix (D = 100, K = 600)



From Figure 7, we can see that, our classifier performed slightly poor for class 4 (International) and class 5 (Science and Tech.). We observe that those classes are often confused with class 6 (Economics). One possible reason for that is, Economic category documents often contains common words and topics of both the Science & Tech. and International news. This is a possible reason for which our system achieved F1-score < 95%. Another reason for which we think our system could perform better is the size of the data for word embedding training. The more data we use to train, the more accurate the vector representations will be, therefore the quality of clusters. Typically, corpus contains 100 billion words for language like English. Bengali has very few online content compared to that.

VI. CONCLUSION

We demonstrate that Bengali word embeddings can be used to create word clusters that capture the semantic relationship of words from the context. We use the clustering information of words as features to perform NLP task like document classification. We achieve the performance of ~91% as F1-score. We show that we can achieve such a performance without any preprocessing of the Bengali text. It proves the effectiveness of the word embedding model for performing the NLP tasks in Bengali. We observed that the larger the text corpus we use, the better the word clusters can be formed; so we will collect more Bengali data to generate the embeddings. We will continue our study to understand how can we learn the vector representation for each word better by studying other existing embedding models: Polyglot, Gensim and Glove. We will also study the effect of dimension reduction on the document classification. We will use our understanding from those study to solve other classification problem like POS tagging, NER and Sentiment Analysis in Bengali.

VII. ACKNOWLEDGEMENT

This research was partially supported by Search Engine Pipilika, which is a Bengali search engine initially developed by Shahjalal University of Science & Technology (SUST). We thank Pipilika team, specially Mahbubur Rub Talha and Tushar Chakraborty, they provided us newspaper data that greatly assisted the research.

REFERENCES

[1] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of Machine Learning Research 12.Aug (2011): 2493-2537.

[2] Mikolov, T., and J. Dean. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems (2013).

[3] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.

[4] Al-Rfou, Rami, Bryan Perozzi, and Steven Skiena. "Polyglot: Distributed word representations for multilingual nlp." arXiv preprint arXiv:1307.1662 (2013).

[5] Ali, Md Nawab Yousuf, et al. "Morphological analysis of bangla words for universal networking language." Digital Information Management, 2008. ICDIM 2008. Third International Conference on. IEEE, 2008.

[6] Ismail, Sabir, and M. Shahidur Rahman. "Bangla word clustering based on N-gram language model." Electrical Engineering and Information & Communication Technology (ICEEICT), 2014 International Conference on. IEEE, 2014.

[7] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of Machine Learning Research 9.Nov (2008): 2579-2605.

[8] Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern recognition letters 31.8 (2010): 651-666.

[9] Cristianini, Nello, and John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, 2000.

[10] Bengio, Y., Ducharme, R., & Vincent, P. (2001). A neural probabilistic language model. NIPS.

[11] Yao, Kaisheng, et al. "Recurrent neural networks for language understanding." INTERSPEECH. 2013.

[12] Rehurek, Radim, and Petr Sojka. "Software framework for topic modelling with large corpora." In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. 2010.

[13] Chakrabarty, A., & Garain, U. (2016). BenLem (A Bengali Lemmatizer) and Its Role in WSD. ACM Transactions on Asian and Low-Resource Language Information Processing ACM Trans. Asian Low-Resour. Lang. Inf. Process., 15(3), 1-18. doi:10.1145/2835494

[14] P F Brown, P V Desouza, R L Mercer, V J D Pietra, V J Della. and J C Lai. “Class-based N-gram Models of Natural Language”. Computational linguistics, 18 No: 4, 1992, P: 467-479.

[15] Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010.

[16] E E Korkmaz. “A method for improving automatic word categorization”. Doctoral dissertation, Middle East Technical University, 1997

[17] W Ding, H Al-Mubaid and S Kotagiri.“Word classification: An experimental approach with Naïve Bayes”. Conference on Computers and Their Applications, 2009

[18] Chakraborty, Tanmoy, Dipankar Das, and Sivaji Bandyopadhyay. "Identifying Bengali Multiword Expressions using Semantic Clustering." Lingvisticæ Investigationes 37.1 (2014): 106-128.

[19] A. Lopes, R. Pinho, F. V. Paulovich, and R. Minghim, "Visual text mining using association rules," Computers & Graphics, vol. 31, pp. 316-326, 2007.

[20] F. Sebastiani, "Machine learning in automated text categorization," ACM computing surveys (CSUR), vol. 34, pp. 1-47, 2002.

[21] T. Denoeux, "A k-nearest neighbor classification rule based on Dempster-Shafer theory," Systems, Man and Cybernetics, IEEE Transactions on, vol. 25, pp. 804-813, 1995.

[22] Dumais, Susan, et al. "Inductive learning algorithms and representations for text categorization." Proceedings of the seventh international conference on Information and knowledge management. ACM, 1998.

[23] Frank, Eibe, and Remco R. Bouckaert. "Naive bayes for text classification with unbalanced classes." European Conference on Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg, 2006.

[24] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer Berlin Heidelberg, 1998.

[25] Peng, Fuchun, and Dale Schuurmans. "Combining naive Bayes and n-gram language models for text classification." European Conference on Information Retrieval. Springer Berlin Heidelberg, 2003.

[26] Mandal, Ashis Kumar, and Rikta Sen. "Supervised Learning Methods for Bengali Web Document Categorization." arXiv preprint arXiv:1410.2045 (2014).

[27] Chy, Abu Nowshed, Md Hanif Seddiqui, and Sowmitra Das. "Bangla news classification using naive Bayes classifier." Computer and Information Technology (ICCIT), 2013 16th International Conference on. IEEE, 2014.

[28] Kabir, Fasihul, et al. "Bangla text document categorization using Stochastic Gradient Descent (SGD) classifier." Cognitive Computing and Information Processing (CCIP), 2015 International Conference on. IEEE, 2015.

[29] Yanhong YUAN, Liming HE, Li PENG. "A New Study Based on Word2vec and Cluster for Document Categorization". Zhixing HUANG, China, Journal of Computational Information Systems 10: 21 (2014) 9301–9308

[30] Tong, Simon, and Daphne Koller. "Support vector machine active learning with applications to text classification." Journal of machine learning research 2.Nov (2001): 45-66.

[31] Lindén, Krister, and Jussi Olavi Piitulainen. "Discovering synonyms and other related words." Proceedings of COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology. 2004

[32] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, K. Miller. 1990. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235–244.

[33] S. Benoît, F. Darja. 2008. Building a free French wordnet from multilingual resources. In Proc. of Ontolex 2008, Marrakech, Maroc.

[34] Pushpak Bhattacharyya, IndoWordNet, Lexical Resources Engineering Conference 2010 (LREC 2010), Malta, May, 2010.

[35] E. Voorhees and D. Harman, TREC: Experiment and Evaluation in Information Retrieval. MIT Press Cambridge, 2005, vol. 63.