Sliding Window Based Text Categorization using TF_IDF and Word2Vec Features

Mrs.V.Kumuthavalli, M.C.A., M.Phil.,
(Regno: 8143 Research Centre: St.Xavier's College, Palayamkottai)
Associate Professor, Department of Computer Science, Sri Parasakthi College for Women, Courtallam – 627 802.
(An Autonomous College of ManonmaniamSundaranar University, Abishekapatti, Tirunelveli–12, Tamil Nadu, India)

Dr.V.Vallimayil, M.C.A., M.Phil., Ph.D.,
Principal, Palanisamy College of Arts, Perundurai–638 052,
Erode (D.T), Tamil Nadu, India.

Abstract: Text mining is termed as extraction of relevant yet hidden information from the text document. Through the sudden growth in Digital World, the task of organizing text data becomes one of the principal problems. With the rapid growth of online information, there has been an explosion in the volume of documents. The purpose of document classification is to allocate the contents of a text or document for one or more categories. It is employed in document association and management, information retrieval, and certain machine learning algorithms. The main objective of this paper is to design a text categorization algorithm on the basis of sliding window using Term Frequency and Inverse Document Frequency along with Word2Vec features. The performance of the proposed system is analyzed with the help of NewGroup20 dataset with Decision Tree, Naïve Bayes, Random Forest and K-nearest neighbor classification algorithms.

***Index Terms*–Text mining, Word2Vec, TF_IDF, Sliding Window, Classification.**

I. INTRODUCTION

The goal of text categorization is the classification of documents into a fixed number of predefined categories. Each document can be in multiple, exactly one, or no category at all. Using machine learning, the objective is to learn classifiers from examples which perform the category assignments automatically. This is a supervised learning problem. Since categories may overlap, each category is treated as a separate binary classification problem.

The first step in text categorization is to transform documents, which typically are string of characters, into a representation suitable for the learning algorithm and the classification task. Information Retrieval research suggests that word stems work well as representation units and that their ordering in a document is of minor importance for many tasks. This leads to an attribute value representation of text. Each distinct word ω_i corresponds to a feature with the number of times word $\omega_i$occurs in the document as its value. To avoid unnecessarily large feature vectors, words are considered as features only if they occur in the training data at least 3 times and if they are not "stop-words"(like "and", "or", "it", etc.). From IR, it is known that scaling the dimensions of the feature vector with their inverse document frequency (IDF) improves performance.

Along with the rapid and fast development of Internet, there is a tremendous increase in the use of online data and information. The exponential growth of data has led us to an information explosion era, where the data cannot be easily maintained. A large amount of information is present on the social media which is analyzed with the help of text mining to generate meaningful patterns and latest trends. A lot of work has been done on text categorization, In order to improve the performance of classification this paper combines the strength of sliding window analysis along with weighted term based features.

Text classification methods are applied to a dataset of NEWSGROUP20. More specifically, we compare four supervised machine learning algorithms: Naive Bayes (NB), K-Nearest Neighbours (KNN-IBK), Decision Tree and Random Forest for document classification. The measured results of our experiment shows that the Naïve bayes algorithm outperforms other algorithms, and that it reaches significant improvement in accuracy compared to other classifier techniques.

II. RELATED WORK

Text Mining or knowledge discovery from text (KDT) − first introduced by Fledman et al. [25] − refers to the process of extracting high quality of information from text (i.e. structured such as RDBMS data [20, 26], semi-structured such as XML and JSON [18, 27, 28], and unstructured text resources such as word documents, videos, and images). It widely covers a large set of related topics and algorithms for analyzing text, spanning various communities, including information retrieval, natural language processing, data mining, machine learning many application domains web and biomedical sciences.

Information Retrieval (IR): Information Retrieval is the activity of finding information resources (usually documents) from a collection of unstructured data sets that satisfies the information need [7, 8]. Therefore information retrieval mostly focused on facilitating information access rather than analyzing information and finding hidden patterns, which is the main purpose of text mining. Information retrieval has less priority on processing or transformation of text whereas text mining can be considered as going beyond information access to further aid users to analyze and understand information and ease the decision making.

Natural Language Processing is sub-field of computer science, artificial intelligence and linguistics which aim at understanding of natural language using computers [12, 6]. Many of the text mining algorithms extensively make use of NLP techniques, such as part of speech tagging (POG), syntactic parsing and other types of linguistic analysis (see [2, 19] for more Information).

Information Extraction is the task of automatically extracting information or facts from unstructured or semi-structured documents [13, 30]. It usually serves as a starting point for other text mining algorithms. For example extraction entities, Name Entity Recognition (NER), and their relations from text can give us useful semantic information.

Text Summarization: Many text mining applications need to summarize the text documents in order to get a concise overview of a large document or a collection of documents on a topic [1, 10].

Unsupervised learning methods are techniques trying to find hidden structure out of unlabelled data. They do not need any training phase, therefore can be applied to any text data without manual effort. Clustering and topic modeling are the two commonly used unsupervised learning algorithms used in the context of text data.

Supervised learning methods are machine learning techniques pertaining to infer a function or learn a classifier from the training data in order to perform predictions on unseen data. There is a broad range of supervised methods such as nearest neighbor classifiers, decision trees, rule-based classifiers and probabilistic classifiers [33, 17].

There are various probabilistic techniques including unsupervised topic models such as probabilistic Latent semantic analysis (pLSA) [35] and Latent Dirichlet Allocation (LDA) [9], and supervised learning methods such as conditional random fields [14] that can be used regularly in the context of text mining.

There are many different applications on the web which generate tremendous amount of streams of text data. News stream applications and aggregators such as Reuters and Google news generate huge amount of text streams which provides an invaluable source of information to mine. Social networks, particularly Face book and Twitter create large volumes of text data continuously. They provide a platform that allows users to freely express themselves in a wide range of topics. The dynamic nature of social networks makes the process of text mining difficult which needs special ability to handle poor and non-standard language [22, 5].

Biomedical Text Mining: Biomedical text mining refers to the task of text mining on text of biomedical sciences domains. The role of text mining in biomedical domain is twofold, it enables the biomedical researchers to efficiently and effectively access and extract the knowledge out of the massive volumes of data and also facilitates and boosts up biomedical discovery by augmenting the mining of other biomedical data such as genome sequences and protein structures [15].

With the advent of e-commerce and online shopping, a huge amount of text is created and continues to grow about different product reviews or users opinions. By mining such data, we find important information and opinion about a topic which is significantly fundamental in advertising and online marketing [4].

Bo Tang, Student Member, IEEE, Steven Kay, Fellow, IEEE, and Haibo He, Senior Member, IEEE: Feb 2016, Proposed [16] Automated element determination is imperative for text arrangement to lessen highlight survey and to speed learning procedure of classifiers. Kapila Rani, Satvika, Proposed [27] Text classification can be quickly depicted as the automatization of the report association procedure to an arrangement of precharacterized classifications. Programmed Text Classification is an essential application for the distinguishing proof of computerized reports.

Naive Bayes Classifiers are simple probabilistic classifiers based on the Bayes Theorem [29]. These are highly scalable classifiers involves a family of algorithms based on a common principle assuming that the value of a particular feature is independent of the value of any other feature, given the class variable. Data pre-processing reduces the size of the input text documents significantly. It involves activities like sentence boundary determination [21], natural language specific stop-word elimination [26] [28] [31] and stemming [28] [24].

For English language, the Porter’s stemmer is a popular algorithm [30] [3], which is a suffix stripping, sequence of systematic steps for stemming an English word, reducing the vocabulary of the training text by approximately one-third of its original size [30]. In cases where the source documents are web pages, additional pre-processing is required to remove / modify HTML and other script tags [32]. Feature extraction / selection helps identify important words in a text document. This is done using methods like TF-IDF (term frequency-inverse document frequency) [11], LSI (latent semantic indexing) [34], multi-word [28][23] etc.

III. Methodology

The proposed system is designed to categorize the text documents. It consists of two phases one as training and another one as testing phase. In both sections the following process are carried out. Such as pre-processing, sliding window construction and feature extraction. The block diagram of the proposed system is shown in the figure 1.

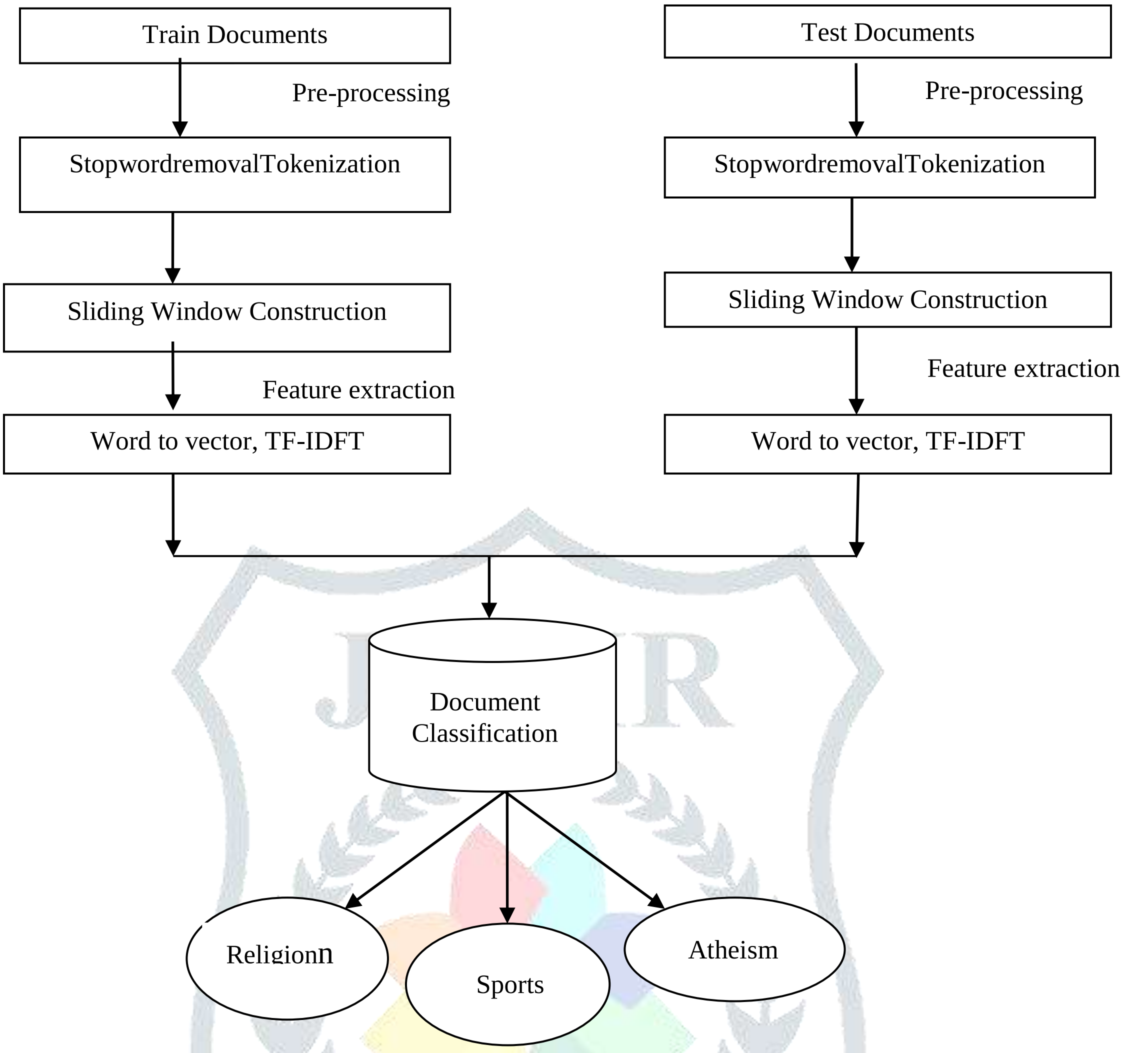


Figure 1 Architecture of the proposed system.

3.1 Preprocessing

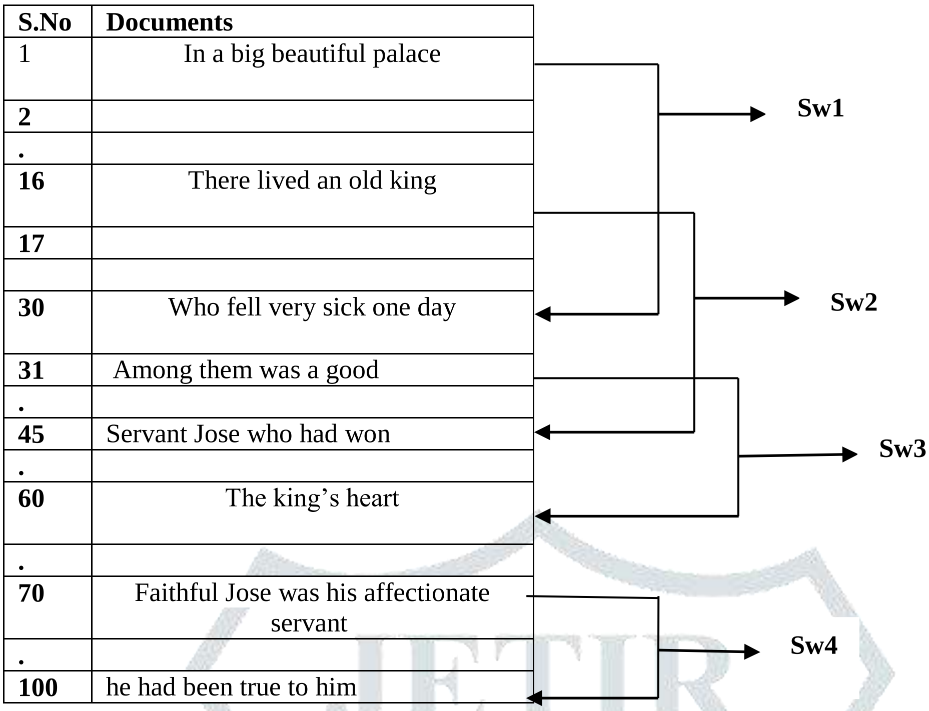
It involves all processes, methods that are required to prepare data for text mining. It converts data from original form to machine readable format before applying feature extraction methods to generate new collection of documents represented by the concepts. Techniques like stop word removal; stemming and tokenization are involved in preprocessing.

□**Stop word removal**: Very often a common word, which would appear to be less significant in selecting document that would match a user's need, is completely expelled from the vocabulary. Such words are called stop words and the technique is called stop words removal technique. This technique increases the effectiveness and efficiency. Example of the stop words area, is, the, when, etc.

□**Tokenization**: Tokenization is the process of breaking up given character sequences into meaningful words, symbols, or crunches of data while maintaining its security and integrity which can be further used for processing.

3.2 Sliding Window Construction

Sliding Window concept is mostly used for time series analysis. In those cases it partitioned the data into number of sectors with same dimension, and use those sub data for further process. Normally in the existing works the features are extracted from the whole document using TF-IDF or word2vec model. In this proposed model the document is analyzed part by part using sliding window concept. Due to this process the term weights are calculated on the basis of location. Hence this feature extraction improves the classification accuracy in an effective manner. The following figure 2. shows an example of sliding window concept with window size of 30% and overlapping window size of 15% of a document of size (N=100) lines.



S.No	Documents
1	In a big beautiful palace
2	
.	
16	There lived an old king
17	
30	Who fell very sick one day
31	Among them was a good
.	
45	Servant Jose who had won
.	
60	The king's heart
.	
70	Faithful Jose was his affectionate servant
.	
100	he had been true to him

Figure 2 Sliding Window Construction Model

3.3 Feature Extraction

3.3.1 Word Embeddings:

The core idea behind word embedding is to assign such a dense and low-dimensional vector representation to each word that semantically similar words are close to each other in the vector space. The merit of word embedding is that the semantic similarity between two words can be conveniently evaluated based on the cosine similarity measure between corresponding vector representations of the two words. In the popular word embedding word2vec a two layer neural network language model is designed to learn vector representations for each word.

To prepare the dataset for learning involves transforming the data by using the StringToWordVector filter, which is the main tool for text analysis in PYTHON. The StringToWordVector filter makes the attribute value in the transformed datasets Positive or Negative for all single-words, depending on whether the word appears in the document or not. This filtration process is used for configuring the different steps of the term extraction.

3.3.2 Inverse Document Frequency (IDF):

Sets whether if the word frequencies in a document should be transformed into fij×log (no of document with word i) where fij is the frequency of word i in document (instance) j.

3.3.3 Term frequency (TF): Sets whether if the word frequencies in a document should be transformed into log (1+fij) where fij is the frequency of word i in document (instance) j.

$$\mathbf{tf}\ (t, d) = \mathbf{log}\ (1 + f_{t,d}) \qquad (1)$$

3.3.4 TF-IDF: TF-IDF score is the product of the Term Frequency (TF) of the term in that document and the Inverse Document Frequency (IDF) of that term in the corpus. Mathematically,
TF-IDF can be denoted by,

$$\mathbf{tfidf\ (t, d, D) = tf\ (t, d).idf\ (t, D)} \qquad (2)$$

3.4 Algorithm of the proposed model

Following steps are performed for conduction of the experiment.
Step 1: Feed the newsgroup dataset to PYTHON TOOL.

Step 2: Remove the Stop words from the dataset (i.e. is, the, on.etc.)

Step 3: Do stemming word analysis for removing the commoner morphological and in flexional endings from the words.

Step 4: Apply Sliding Window concept to split each document into sub documents

Step 5: Calculate the term frequency (TF) and inverse term frequency (IDF) each sub documents

Step 6: Convert the document in text format to Word2Vec representation

Step 7: Extract features based on the TF and IDF and generate feature vector along with word2vec.

Step 8: Categorize the given Test document into particular target class using classifier algorithms such as Random Forest, Naive Bayes, Decision Tree and K Nearest Neighbour.

IV. Experimental Result and Analysis

A large collection of text documents is considered as unstructured data. It is very difficult to group the text documents. A dataset is used for the clustering of documents. For this purpose 20 News Group Dataset is used. The dataset consists of collection of large number of documents. It consists of a collection of 20000 documents partitioned into 20 different categories. In this experiment the following 8 categories are used for analysis such as alt.atheism, misc.forsale, rec.motorcycles, rec.sport.hockey, sci.space, sci.electronics, soc.religion.christian, talk.politics.mideast. Pre-processing techniques are applied to the dataset in order to obtain the pre-processed dataset. Tokenization and stop word removal techniques are applied.

4.1 Results of Performance measures

The performance of the proposed sliding window approach is evaluated using the parameters such as Accuracy, precision, Recall, F1score.

True Positives (TP) - These are the correctly predicted positive values which mean that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP) – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN) – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$\textbf{Accuracy} = \mathbf{TP+TN/(TP+FP+FN+TN)} \qquad (3)$$

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\textbf{Precision} = \mathbf{TP/(TP+FP)} \qquad (4)$$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

$$\textbf{Recall} = \mathbf{TP/(TP+FN)} \qquad (5)$$

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

$$\textbf{F1 Score} = \mathbf{2*(Recall * Precision) / (Recall + Precision)} \qquad (6)$$

The following table 4.1 shows the performance the proposed approach with different classifiers such as Ranfom Forest, Naive Bayes, Decision Tree, and K-Nearest Neighbour.

Table 4.1: Performance measures of proposed methods

CATEGORIES	ACCURACY							
	Word2vec,TF-IDF With sliding window				Word2vec, TF-IDF Without sliding window			
	RF	BAYES	DT	KNN	RF	BAYES	DT	KNN
alt.atheism	0.915	0.9774	0.87875	0.53875	0.652406	0.9700	0.875	0.85625
misc.forsale	0.90625	0.9637	0.875	0.8	0.743316	0.9325	0.94375	0.91125
rec.motorcycles	0.97375	0.9874	0.94125	0.9125	0.754011	0.9850	0.94625	0.96
rec.sport.hockey	0.96875	0.9887	0.895	0.9	0.823529	0.9812	0.90375	0.90375
sci.space	0.90125	0.9649	0.4575	0.90125	0.778075	0.9400	0.875	0.90875
sci.electronics	0.9425	0.9887	0.92375	0.8875	0.735294	0.9750	0.92625	0.93625
soc.religion.christian	0.72	0.9824	0.8975	0.8775	0.747326	0.9700	0.41875	0.94
talk.politics.mideast	0.9225	0.9887	0.91875	0.905	0.870321	0.9812	0.89875	0.94375

Analysis on the metrics such as precision, recall and F1-score are shown in the corresponding table 4.2, table 4.3 and table 4.4

Table 4.2: Analysis on the metrics of precision

CATEGORIES	PRECISION							
	Word2vec,TF-IDF With sliding window				Word2vec, TF-IDF Without sliding window			
	RF	BAYES	DT	KNN	RF	BAYES	DT	KNN
alt.atheism	0.775862	0.92	0.615385	0.206972	0.10728	0.97	0	0.460317
misc.forsale	0.578616	0.84	0	0.325581	0.010638	0.67	0.936508	0.626087
rec.motorcycles	0.934066	0.92	0.949153	0.875	0	0.94	0.938462	0.854167
rec.sport.hockey	0.837838	0.95	0.833333	0.884615	0.1	0.93	0.870968	0.587786
sci.space	0.956522	0.88	0.180077	0.705882	0.064935	0.88	0	0.787234
sci.electronics	0.78125	0.95	0.782609	0.916667	0.009709	0.94	0.825397	0.865672
soc.religion.christian	0.677419	0.93	0.618421	0.6	0	0.83	0.173524	0.842105
talk.politics.mideast	0.637681	0.96	0.972973	0.9	0	0.91	1	0.848101

Table 4.3: Analysis on the metrics of Recall

CATEGORIES	RECALL							
	Word2vec,TF-IDF With sliding window				Word2vec, TF-IDF Without sliding window			
	RF	BAYES	DT	KNN	RF	BAYES	DT	KNN
alt.atheism	0.45	0.9	0.08	0.95	0.509091	0.78	0	0.87
misc.forsale	0.92	0.87	0	0.56	0.01	0.94	0.59	0.72
rec.motorcycles	0.85	0.98	0.56	0.35	0	0.93	0.61	0.82
rec.sport.hockey	0.93	0.96	0.2	0.23	0.04	0.93	0.27	0.77
sci.space	0.22	0.83	0.94	0.36	0.050505	0.63	0	0.37
sci.electronics	0.75	0.95	0.54	0.11	0.010309	0.85	0.52	0.58
soc.religion.christian	0.84	0.93	0.47	0.06	0	0.95	0.97	0.64
talk.politics.mideast	0.88	0.94	0.36	0.27	0	0.94	0.19	0.67

Table 4.2: Analysis on the metrics of F1score

CATEGORIES	FSCORE							
	Word2vec,TF-IDF With sliding window				Word2vec, TF-IDF Without sliding window			
	RF	BAYES	DT	KNN	RF	BAYES	DT	KNN
alt.atheism	0.56962	0.91	0.141593	0.339893	0.177215	0.87	**0**	0.602076
misc.forsale	0.710425	0.86	**0**	0.411765	0.010309	0.78	0.723926	0.669767
rec.motorcycles	0.890052	0.95	0.704403	0.5	0	0.93	0.739394	0.836735
rec.sport.hockey	0.881517	0.96	0.322581	0.365079	0.057143	0.93	0.412214	0.666667
sci.space	0.357724	0.86	0.302251	0.476821	0.056818	0.73	0	0.503401
sci.electronics	0.765306	0.95	0.639053	0.196429	0.01	0.89	0.638037	0.694611
soc.religion.christian	0.75	0.93	0.534091	0.109091	0	0.88	0.294385	0.727273
talk.politics.mideast	0.739496	0.95	0.525547	0.415385	0	0.93	0.319328	0.748603

Compared to all classifiers, the Naïve bayes classifier performs well. The category wise performance metrics such as precision, recall and F1-score for the Naïve bayes classifier is shown in the following figure 3,figure 4 and figure 5.

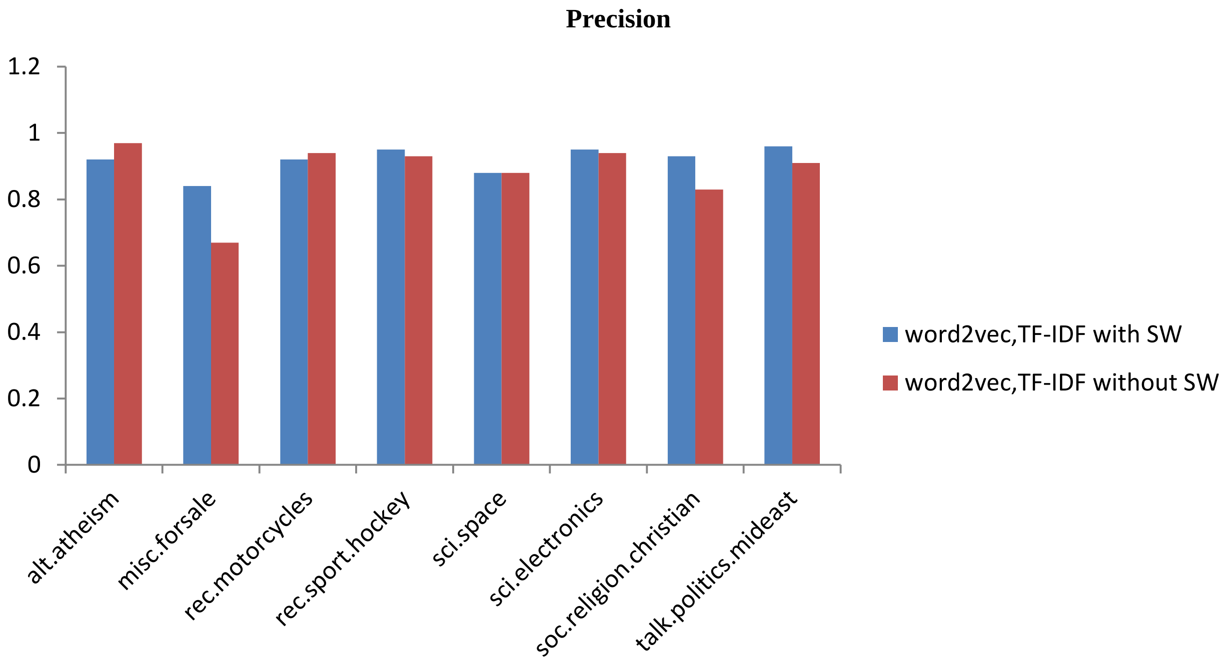


Figure 3 Category wise Precision of Naïve bayes Classifier

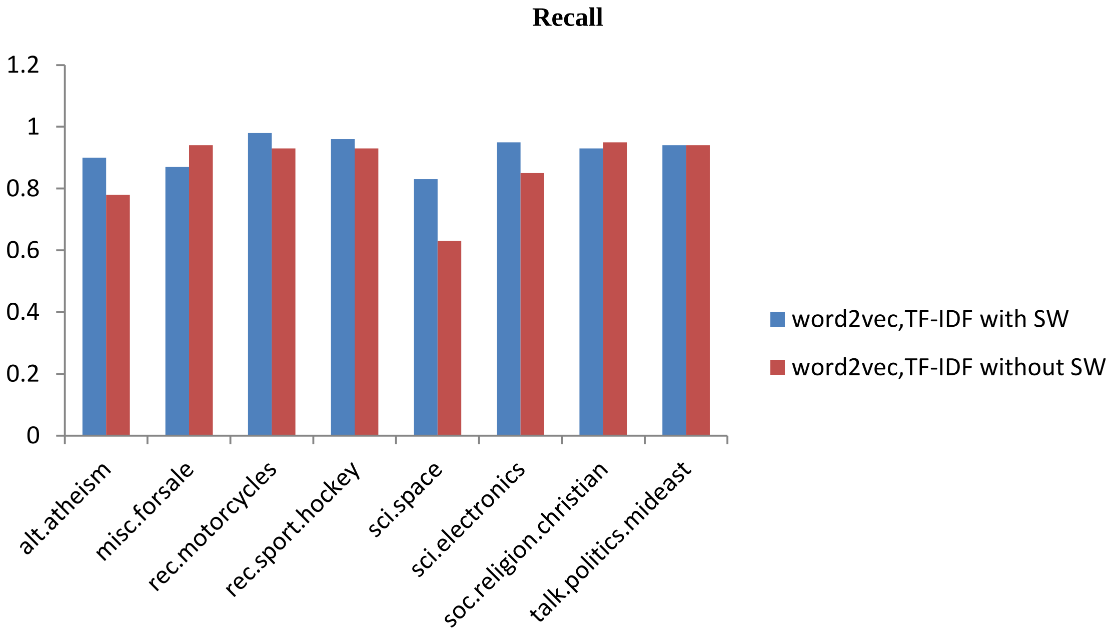


Figure 4 Category wise Recall of Naïve bayes Classifier

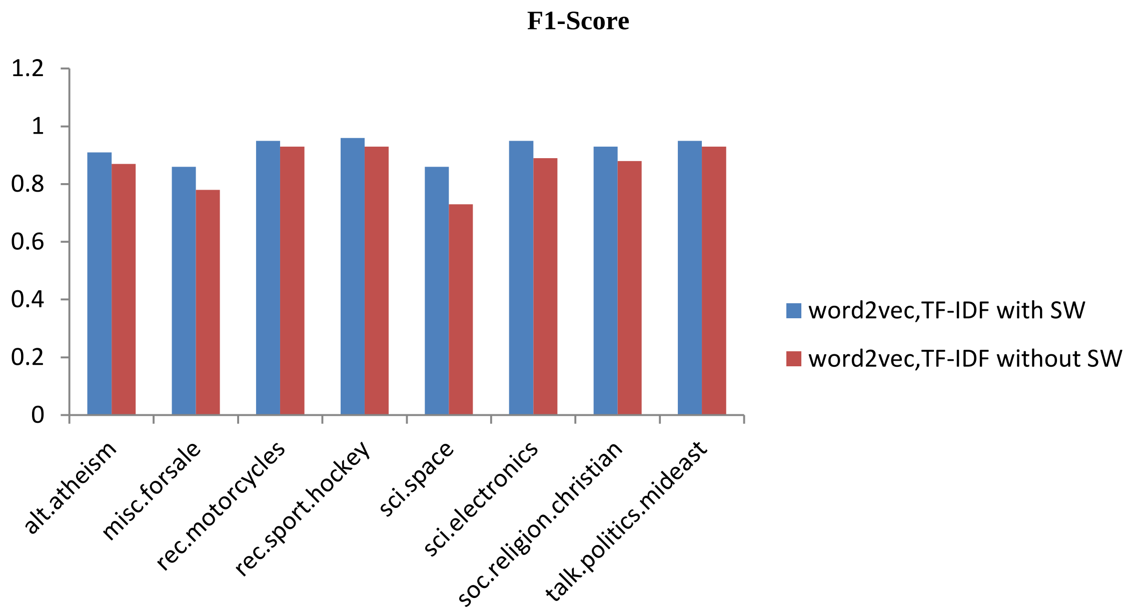


Figure 5 Category wise F1-scorel of Naïve bayes Classifier

V. Conclusion

In this paper the concept of sliding window is used to represent the document in partitioned way. The TF-IDF and Word2Vec features extracted from partitioned document. These kinds of TF-IDF and Word2Vec score improves the class specific representation compared to whole document score. In this paper four kinds of classifiers are used to for classification analysis. The results show that the classifier Naïve bayes provides some significant improvement in accuracy compared to the existing system. In future this concept can be extended to further improve the accuracy of this model by working on deep learning concept.

References

[1] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. 2005. A Brief Survey of Text Mining. In Ldv Forum, Vol. 20. 1962
[2]Anne Kao and Stephen R Poteet. 2007. Natural language processing and text mining. Springer.
[3]BalahurA. and MontoyoA. 2008. "A feature dependent method for opinion mining and classification", In proceedings of the IEEE international conference on Natural Language Processing and Knowledge Engineering, pp. 17.
[4] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. Foundations and trends in information retrieval 2, 1-2 (2008), 1–135.
[5] Christopher C Yang, Haodong Yang, Ling Jiang, and Mi. Zhang. 2012. Social media mining for drug safety signal detection. In Proceedings of the 2012 international workshop on Smart health and wellbeing. ACM, 33–40.
[6] Christopher D Manning, HinrichSchütze, et al. 1999. Foundations of statistical natural language processing.Vol. 999. MIT Press.
[7] Christopher D Manning, PrabhakarRaghavan, and HinrichSchütze. 2008. Introductionto information retrieval. Vol. 1. Cambridge University press Cambridge.
[8] Christos Faloutsos and Douglas W Oard. 1998. A survey of information retrieval and filtering methods. Technical Report.
[9]DavidMBlei, AndrewY Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. The Journal of machine Learning research 3 (2003), 993–1022.
[10]Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. Computational linguistics 28, 4 (2002), 399–408.
[**11]** Durant K. T., Smith M. D., 2006. "Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection".
[12] Elizabeth D Liddy. 2001. Natural language processing. (2001).
[13]FabrizioSebastiani. 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR) 34, 1 (2002), 1–47.
[14] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional Random fields: Probabilistic models for segmenting and labeling sequence data.
[15] Juan B Gutierrez, Mary R Galinski, Stephen Cantrell, and Eberhard O Voit. 2015. From within host dynamics to the epidemiology of infectious disease: scientific overview and challenges.

[16]Jiawei Han and MichelineKamber. 2011. “Data Mining Concepts and Techniques”, Morgan Kaufman publishers, San Francisco, Elsevier, pp. 285-351.
[17] Jim Cowie and Wendy Lehnert. 1996. Information extraction. Common. ACM 39, 1 (1996), 80–91.
[18]MahmoodDoroodchi, AzadehIranmehr, and Seyed Amin Pouriyeh. 2009. An investigation on integrating XML-based security into Web services. In GCC Conference & Exhibition, 2009.5th IEEE. IEEE, 1–5.
[19] Martin Rajman and RomaricBesançon. 1998. Text mining: natural language techniques and text mining applications. In Data Mining and Reverse Engineering. Springer, 50–64.
[20] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. 1996. Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and data Engineering8, 6 (1996), 866–883.
[21]M.Sukanya, S.Biruntha. "Techniques on Text Mining”, International Conference on Advanced Communication Control and Computing Technologies, IEEE-2012
[22]Nidhi, Vishal Gupta. December 2011. "Recent Trends in Text Classification Techniques", International Journal of Computer Applications (0975 – 8887) Volume 35– No.6.
[23]Polpinij J. and Ghose A. K. 2008. “An ontology-based sentiment classification methodology for online consumer reviews”. In proceedings of the IEEE international conference on Web Intelligence and Intelligent Agent Technology, pp. 518-524.
[24]PritamGundecha and Huan Liu. 2012. Mining social media: a brief introduction. In New Directions in Informatics, Optimization, Logistics, and Production. Informs, 1–17.
[25] Ronen Feldman and Ido Dagan. 1995. Knowledge Discovery in Textual Databases (KDT). In KDD, Vol. 95.112–117.
[26]SašoDžeroski. 2009. Relational data mining. In Data Mining and Knowledge Discovery Handbook. Springer, 887–911.
[27] Seyed Amin Pouriyeh and MahmoodDoroodchi. 2009. Secure SMS Banking Based On Web Services. In SWWS.79–83.
[28]Seyed Amin Pouriyeh, MahmoodDoroodchi, and MR Rezaeinejad. 2010. Secure Mobile Approaches Using Web Services. In SWWS.75–78.
[29] S. Subbaiah, "Extracting Knowledge using Probabilistic Classifier for Text Mining", Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, February 21-22, IEEE-2013.
[30]SunitaSarawagi et al. 2008. Information extraction. Foundations and Trends in Databases 1, 3 (2008), 261–377.
[31]Sonali Vijay Gaikwad, ArchanaChaugule, PramodPatil. January 2014. "Text Mining Methods and Techniques, "International Journal of Computer Applications (0975 – 8887) Volume 85 – No 17.
[32] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In Proceedings of The 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 50–57.
[33] Tom M Mitchell. 1997. Machine learning. 1997. Burr Ridge, IL: McGraw Hill 45 (1997).
[34]WebKDD. 2006. LNAI 4811, pp. 187-206, Springer-Verlag Berlin Heidelberg.
[35] Zhao L., and Li C... 2009. “Ontology based opinion mining for movie reviews”, KSEM 2009, LNAI 5914, pp. 204-214, Springer-Verlag Berlin Heidelberg.

.