I.J. Modern Education and Computer Science, **2018, 9, 11-19**
Published Online September 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2018.09.02

Developing an Efficient Text Pre-Processing Method with Sparse Generative Naive Bayes for Text Mining

Mrutyunjaya Panda
Department of Computer Science and Applications, Utkal University, Vani Vihar, Bhubaneswar-4, India
Email: mrutyunjaya74@outlook.com

Received: 03 June 2018; Accepted: 14 August 2018; Published: 08 September 2018

Abstract—With the explosive growth of internet, there are a big amount of data being collected in terms of text document, that attracts many researchers in text mining. Traditional data mining methods are found to be trapped while dealing with the scale of text data. Such large scale data can be handled by using parallel computing frameworks such as: Hadoop and MapRedue etc. However, they are also not away from challenges.On the other hand, Naive Bayes (NB) and its variant Multinomial Naive Bayes (MNB) plays an important role in text mining for their simplicity and robustness but if anything or everything from number of words, documents and labels go beyond the linear scaling, then MNB is intractable and will soon be out of memory while dealing in a single computer. Looking into the high dimensional sparse nature of the documents in text datasets, a scalable sparse generative Naive Bayes (SGNB) classifier is also proposed to develop a good text classification model. Unlike parallelization, SGNB reduces the time complexity non-linearly and hence expected to provide best results. In this paper, an efficient Lovins stemmer in combination with snowball based stopword calculation and word tokenizer is proposed for text pre-processing. The extensive experiments conducted on publicly available very well known text datasets opines the effectiveness of the proposed approach in terms of accuracy, F-score and time in comparison to many baseline methods available in the recent literature.

Index Terms—Information Retrieval, Stemming, Tokenization, stop-word, Sparse Generative classifier, Naive Bayes, Accuracy, W-T-L

I. Introduction

Information retrieval (IR) [1] referred to as current science of searching for documents, meta data about the documents from the internet and extracting the information from them. The example include but not limited to: Google, PUBMED central and Alta vista etc. Due to the huge access of internet, digital world enables us to collect lots of digital information in a text database. Text mining seems to be a branch of Data mining in which unstructured or semi-structured text data are used for discovery of the knowledge hidden within the text. Due to its complexity in processing, it has attracted many a researchers for carrying out further research in this field. The most important job in the text mining is text classification, where word of text are considered to be the features. Text categorization, at the same time is also another methodology in text mining frequently used as a part natural language processing system, used to assign subject categories to the text documents and to filter text, for better classification results [2].

Information retrieval (IR) is centered with prime objective of automatically analyzing the documents for extraction of valuable information as per the need of the user. This process may be thought of in two steps that includes: Tokenization as a process where character streams are chopped into tokens and then linguistic pre-processing methods where equivalence of classes tokens as a set of terms that are indexed. Linguistic pre-processing methods include Stemming, lemmatization and stop word removal to name a few.

In the context of generally speaking English text, individual English words are called as a token. More specifically, token may be an instance of character sequence in a document. Tokenization is a process of adding white space or throwing away of punctuation characters or else using some non-alphanumerical characters as delimiters between the token. In this, a "type" is a class of all token containing the same sequence of characters where as "term" is a normalized type that is indexed in the IR system directory. The example of a tokenization process is illustrated below:

- Input: "Doctors, students, Countrymen, give us your best"
- Output: |Doctors|students|Countrymen|give|us|your|best|

At the same time, the pre-processing by removal stop words from the document where semantically non-selective words are excluded from the dictionary in entirety. Here, there are two principal strategies are followed like: the terms are sorted in terms of their frequency of occurrence and then, probably by adding the most frequent ones to the stop list. Earlier IR systems, a

quite large stop lists with 200-300 terms were available but, today's trend is to have a small stop list with 7-12 terms or else no stop list at all as found in web search engines. An example of stop list in Reusters-RCV1 dataset are as follows:

a	and	as	at	be	by	for	from	has	he
in	is	it	its	of	on	s	said	that	the
to	was	will	with	year					

If the searching of a phrase is done, then stop lists may have some adverse effects on the system's performance.

In IR, information not only addresses direct questions (i.e. what is stemming?), but also to perform a document search for obtaining terms or set of terms inside the document (otherwise known as stemming) or else to complete an abstract search to find an ambiguous word (computer program to search the root of the words, in this case). Hence, a stemming program (or a stemmer) with its evolution in computer science since 1960's, is intended to obtain the stem of a word i.e. morphological root by removing the affixes containing the grammatical and lexical information about the word [4]. In some cases, it is always sufficient to see that related words falls in a same stem even though the stem itself is not in a valid root. It is worth noting here that the words with the same stem are synonymous, to most of the search engines through a process of query expansion called conflation. For the root word "cat", the English Stemmer may identify strings "catlike", "catwalk" etc., however the words like, "important" and "Imported", reduce to the stem "import". Here, the stem is neither a word nor a root in itself.

The large features available in text classification demands suitable feature selection techniques to address the curse of dimensionality problem without sacrificing the essence of the original text data [3]. Apart from the high dimensionality problem, the text mining is characterized to have class imbalance problem with high class skew. In this scenario, with 1% of the training dataset of positive class, the classifier can provide 99% classification accuracy by predicting the rest of the cases for the negative class.

In text mining, it is also observed that the training data is very small where the person's judgment is needed to determine the interest level or topic level and hence, there are always a chance of having very few common word and very large rare words in the dataset, which poses challenges in classification task.

The goal of this paper is to put some inside on the effectiveness of tokenization, stemming and stop word removal strategy for making a good information retrieval process suitable for better text classification.

Naive Bayes classifier is one of the most popularly used classifier in text mining applications for its simplicity yet make strong independence probabilistic assumptions. Further, it is computationally intensive both in terms of memory and CPU usage and importantly, takes a small amount of training data to make useful predictions. It is also opined that the Naive Bayes classifier outperforms many well established classifier such as: AdaBoost, Random forest and Support vector machines etc. [5] .

Even though it has many attractiveness, still there are some issues to be addressed by using Naive Bayes (NB). It is also proved [6] that despite providing low quality probability estimates and hence over estimating the probability of selected class thereto, Naive Bayes still considered to be an most accurate classifier for decision making until we are not intended to predict actual probabilities most accurately.

The main motivation in using the NB is due to its efficiency, however, the conditional independence assumption do not hold good for text classification. Also, it does not provide best accuracy over the others, but it may be well suited in applications where many important attributes of the datasets combinedly produces a best result. More importantly, it is robust to noisy attributes and deals with ease for data contains some gradual change process as a concept drift situations. Multinomial Naive Bayes (MNB), an improvement over NB is another option to deal with text mining, but it may be intractable if the text documents are nonlinear in scale.

Hence, a variant of NB such as Sparse generative Naive Byes as a scalable ones, are proposed to check the effectiveness of the classifier when there are large amount of training data are present to make better predictions as opposed to the smaller training data as in the general NB and MNB model cases. At the same time, it is to investigate, whether this variant can effectively address the robustness of the classifier to concept drift situations in text mining application in a better way or not.

In this paper, a novel Iterated Lovins Stemmer, Rainbow Stop words and Word Tokenizer based Information Retrieval approach in combination with sparse generative Naive Bayes classification model is proposed for developing an efficient text mining model.

Motivation

The motivation behind exploring the effectiveness of the information retrieval based Sparse generative Naive Bayes model for text classification is to perform sentiment analysis, text mining and/or opinion mining with improvements to move a step forward towards natural language processing research. It aims at not only improve upon the time complexity but also to avoid space complexity as was in the case of generative naive Bayes (NB and MNB).

Objective of the research

The main objective of this research centered at to predict the sentiment of the users in different scenarios. Even though, lots of people tried doing research in this interesting area, the Scalability issues are yet to be explored to its fullest extent. This paper proposes an efficient text pre-processing approach with SGNB classifier to obtain better results in sentiment detection and classification. Our proposed approach consists of four stages: Pre-processing with selection of suitable tokenizer, stemming, a proper stemmer and an efficient

stop word removal method, followed by a novel SGNB classifier for text mining purposes.

Contributions

In this paper, we provide the following contributions:

- Perform the most suitable pre-processing techniques to make the text datasets ready for feature selection, to address the curse of dimensionality.
- Identify the most appropriate feature selection algorithms to select the best features, removing the redundant ones
- Application of novel supervised learning algorithm- SGNB for text classification, in order to address the time complexity of the Naive Bayes models
- Compare the text classification results for the a collection data set with the other state of the art research available in the web.

II. Related Work

In Ref. [7], the author proposes to use a novel unsupervised dependency parsing-based text classification method for sentiment prediction from text messages with tweets and reviews. Sentiment analysis problem comprises of sentiment identification to identify the subjective features in text followed by classification to classify the sentiment as positive, negative or neutral [8]. The authors in Ref. [9] proposes to use Naive Bayes, Max Entropy, and Support Vector Machine algorithms on twitter data streams and then provided a good survey on machine learning based and lexicon based approach for sentiment analysis with evaluation metric, its challenges and possible solutions.

Three- way model such as: Uni-gram model, a feature based model and a tree kernel based model is developed by [10] for sentiments analysis. They concluded that the tree kernel based method performs better than the other two, saying that combination of part of speech tags (POS) of polarity of words Apriori may play a bigger role in enhancing the performance of the classifier. In Ref. [11], the author uses Uni-gram Naive Bayes model on Twitter data classification after eliminating the irrelevant attributes (or features)using mutual information and chi-square as feature extraction method. K-nearest neighbor along with twitter-user defined hash tags are used for sentiment analysis [12]. In Ref. [13], it is proposed by the authors to use ensemble of classifiers by using fixed combination, weighted combination and meta classifier combination approaches with feature space (Part-of-speech information and Word relations) and classifiers such as: NB, Maximum Entropy and Support Vector Machines). They concluded that their proposed approach generates better accuracy.

The authors [14] advocated for the usefulness of the Naive Bayes for today's large and sparse dataset with lots of missing values. They detailed about the convergence property of the Naive Bayes with their AUC (Area under the Curve) and given an impression about the NB linear behavior in time and space complexity with the size of the non-missing values.

The authors [15] provided good survey on the effectiveness of different Naive Bayes model on the authorship attribution in Arabic text. They concluded that Multi-variant Bernoulli naive Bayes (MBNB) model provides the best accuracy of 97.43% in comparison to all other variants.

A simple dictionary based stop-word removal algorithm for implementation in Sanskrit language [16]. They tested their approach over various Sanskrit corpora available and found that stopword removal improves the indexing, thus accuracy is high and envisioned of getting much better result if segmentation of Sanskrit word shall be followed in future.

In Ref. [17], the authors used wordnet with word sense disambiguation techniques applied into neuters 21578 and 20 Newsgroup datasets to determine the correct sense, but concluded that their approach need to be improved further for proper identification of synonym and hyponym sysnsets.

The authors [18] presented a good review on the usefulness of light stemmer with their intention of use in the information retrieval process and finally, discussed their effectiveness in terms of precision and recall.

A recurrent Convolutional neural network for text classification and provided its suitability over the traditional approaches in most of the datasets they used. They used both English and Chinese text datasets for classification with various taxonomy in terms of topic classification,sentiment classification and writing style classification [19]. In Ref. [20], the authors evaluated the performance of ten filter based feature subset selection strategy along with four diverse classifier, applied on high dimensional micro blog tweet datasets for better sentiment classifications. Next, they confirmed by their simulation results and validation by statistical methods that the reduced feature size of 75 to 200 provides enhanced accuracy in comparison to no feature selection at all and if the feature size reduced to 50, then there is no change in the accuracy further.

A novel associative classifier for text classification for obtaining high readability and best accuracy, in comparison other approaches [21]. The authors [22] proposed genetic algorithm based feature selection for high dimensional text dataset classification with F1 score measure using SVM, MaxEnt and SGD classification algorithms for their efficiency.

The rest of the paper is organized as follows. Section 3 describes about the proposed methodology followed by details on dataset used in Section 4. While Section 5 discusses about the experimental setup, the experimental findings are provided in Section 6. Finally, conclusion with future scope of work is presented in Section 7.

III. Proposed Methodology

This section discusses about the methodology proposed in detail.

A. Stages of Information Retrieval

There are three stages of Information retrieval process that includes: tokenization, stemming and stop word removal.

A.1. Tokenization

Being an integral part of the information retrieval process, tokenization is a pre-processing step to generate respective tokens from a given documents. The identified words, numbers and other characters in a text document are called as tokens and the segregation of them results tokenization.

In sentence tokenization, the document texts are separated into individual sentences where as in word tokenization, the texts are broken into chunks of word. The advantages of using tokenization in information retrieval are many fold: first, it can provide a reduced search and second in effective use of reduced storage space [28].

Porter tokenization algorithm is well established method in information retrieval system, but it presents poor accuracy during the token identification [29].

The tokenization process involves in three phases. At first, words are extracted from the document while stop words like: the, as, of, and, or, to etc., and special characters like: @, !, &, %, # etc., which do not play any vital role in information retrieval process are to be removed. This infrequent word and letter removal enhances the effectiveness and efficiency of the process with reduced indexing file size. Then, stemming is applied to further enhance the accuracy of the process followed by frequency count of each word at last.

For example, the text information in a document is like " This is an information retrieval paper and it is popularly used in text mining applications. The name of the project is IRTM. In this project, a team of 5 members present". Now, when this text document is passed through tokenization process, then the output generates with the words and numbers are separated from others and finally, produces their frequency count shown in angular braces < >. this is illustrated below.

Input: " This is an information retrieval paper and it is popularly used in text mining applications. The name of the project is IRTM. In this project, a team of 5 members present"

Output:
Words=
this<2>is<3>an<1>information<1>retrieval<1>paper<1>and<1>it<1>popularly<1>used<1>in<2>text<1>mining<1>applications<1>the<2>name<1>of<2>project<2>IRTM<1>members<1>present<1>
Numbers=5<1>

Since, tokenization process finds the distinct keywords and count their frequency, it plays a vital role in probabilistic information retrieval process for achieving better results.

A.2. Stemming

Stemming is the process of removing prefixes and suffixes from the words, thereby seems to be imperative in the information retrieval process. Further, this is considered to determine the stem of the word with word stem as main component while removing the elements like: tense, case, gender, person etc., those indicate the plurality. For example: Consider a case of searching for a document titled "How to celebrate" with a query "celebration" may result nothing in the search space. Now, with the usage of stemming process in the search query, the "celebration" may be stemmed to "celebrate" for making retrieval process becomes successful. Hence, in information retrieval, the stemming is seen as a pre-processing step which increases the reliable improvement of retrieved documents by 10 to 50 times [23]. It is worth noting here that high precision stemmer is needed while dealing with word order and information carrying affixes such as: part of speech, plurality and tense etc., for development of a sophisticated question/ answer information system.

Even though, there are some limitations pointed out by the authors [23, 24], still Porter stemmer is the most widely used stemming algorithm [25] where a set of rules are applied iteratively to remove the suffixes from the word till no rules remain to be applied further.

The important disadvantage of Porter stemming algorithm lies in complete ignorance of the prefixes, so that "understood" and "misunderstood" are considered to be unrelated tokens. Further, the Porter stemmer can conflate words with different senses or meaning like: "several" becomes "sever" and "continuation" becomes "cont".

The Lovins stemmer [26] is faster and more aggressive than Porter stemmer for stemming English words, with no rule iterations has a larger set of suffixes, where each suffix may include multiple morphemes, for better stemming performance, but still it is not away from drawbacks of over conflation and non-word stem. To compare, it can be found that Lovins stemmer maps two words to the same stem but may result in wrong interpretations like: "neurologist" and "neurology" maps to a same stem "neurolog". At the same time, Porter stemmer map the two words to two different stems as "neurologi" and "neurologist". Further, Porter stemmer maps correctly "Police" and "policy" to "polic" and "polici", but incorrectly stemmed to "polic" by lovins stemmer.

In order to minimize the difficulties encountered by Lovins stemmer, a more aggressive Iterated Lovins stemmer [27] is proposed in this research, where the algorithm applies Lovins stemmer repeatedly till no further changes in word is observed. For given input

"beautiful", Lovins Stemmer generates "beauti" as output in the first time of experiment. Second time, with "beauti" as input, the Lovins Stemmer will produce "beau" as output and the process continues till there is no further changes in the word stem. Hence, for this, the iterated Lovins Stemmer produces "beau" as output for input "beautiful", which increases the aggressiveness of the stemmer.

A.3. Stop word removal

Stop words are those extremely common words such as pro-nouns, articles and prepositions, whose do not help text mining in classification techniques because of its little importance in information retrieval process includes but not limited to: "a", "an", "the", "with", "you", "@", "$", "!" etc. Hence, they are required to be removed from the text entirely before classification step for enhancing the efficiency of the classifiers. A stop list may be obtained from the term frequency by calculating the number of times a term appears in the whole text document, which are then discarded during indexing process. While web search engines do not use stop lists at all, the IR system uses normally varies from no stop lists to 7 to 12 terms in a small stop list and then 200 to 300 terms in a large stop list.

In some cases, like phrase query: "President of India" with one stop word "of" seems to be more meaningful than "President" AND "India". Similarly, for "the train to Bhubaneswar", the meaning is lost once the stop word "to" is removed. Hence, in modern IR system, it is found that stop word inclusion do not have much adverse affect on the performance of the text mining results neither in terms of index size nor in terms of query processing time.

Still, we used rainbow stop word removal approach in this paper. Rainbow is a program which is based on Bow library [28], used for text mining purpose. Rainbow is used to set the word vector weights as per our proposed classification method and smoothing of word probabilities is done as per laplacian method and scoring queries for retrieval followed by classification.

B. Sparse generative Naive Bayes model

There are a lot of good classifiers both base and ensemble ones, being applied in text mining including: Neural Network, Support vector Machine, Naive Bayes, Random Forest, Decision trees and logistic regression etc., where except naive Bayes, all are discriminative classifier. Naive Bayes is the only generative classifier among the above classifiers. The basic difference lies in probability inference structure where generative models are full, joint probabilistic models. Randomly generate observable data values given some hidden parameters and simulate (i.e. *generate*) values of any variable in the model. In contrary, discriminative ones models with conditional probabilities allows only sampling of target attributes conditional to the observed attributes.

The text document usually consists of tens or hundreds of words out of whole lot of possible hundreds of thousands words available, making the word vector extremely sparse. This degree of data sparsity along with dimensional increases as the data size gets bigger. Similarly, the accumulated count numbers corresponding to a label also seems to be sparse and finally, the useful representations of text are high dimensional sparse data.

Naive Bayes being a simple ones, robust in dealing with missing attributes and ability to make faster modeling attracts us to use a variant of this for the proposed research in this paper.

Now, for the problem of text classification at hand, with large text documents containing sequential information in the form of a natural language with sparsity in data and high dimensional in nature, scalable Sparse generative naive Bayes model (SGNB) [29] is proposed.

SGNB uses sparse posterior inference [29] to the Naive Bayes by using sparsity in the parameters to address the time and space complexity in an efficient manner. This uses three techniques for efficient computation such as: Log-domain computation generally recommended for large scale statistical model building, precomputing allows us to use appropriate data structure for less computation while model building and finally, inverted index to obtain the parameters to be updates for information retrieval.

C. Discriminative Multinomial Naive Byes text (DMNBtext)

Discriminative Multinomial Naive Bayes (DMNB) is a variant of Naive Bayes where parameter learning method is applied [30]. The DMNB for its characteristics of combining generative and discriminative learning is found suitable for text classification task.

IV. Datasets Used

This section outlines the details on the datasets used for the experiments.

A. C50 dataset

Reuters C50 [31] is a text dataset which is freely obtained from UCI Machine Learning Repository which contains 2500 documents collected from 50 authors of 50 documents each. All documents are in English with same subtopic for document classification in terms of topic rather than the authors' unique features.

B. The 20 Newsgroups

The 20 newsgroups dataset [32] contains 20000 newsgroup documents which are partitioned equally among 20 different newsgroups. It was originally collected by Ken Lang through his seminal paper. This dataset is extremely popular in text clustering and text classification task using machine learning techniques.

C. Twitter dataset for Arabic sentiment analysis

Twitter dataset [33] consists of 200 labeled tweets falling equally into positive and negative ones, considering topics like: arts and politics etc.. These tweets are written in Arabic and the Jordanian dialect.

This data is oriented towards providing sentiments of the user from the inputted texts.

D. Large Movie review dataset (ACLIMDb v1)

ACLIMDb v1 dataset (IMDb) [34] presents movie reviews with their corresponding sentiment polarity labels. This dataset is often used as a benchmark for sentiment classification in text mining applications. There are 25000 positive and same number of negative reviews present with a condition of not more 30 reviews is allowed for the same movie.

E. Sentence corpus dataset

Sentence Annotation from abstract and introduction of 30 scientific articles are collected in the sentence corpus dataset [35]. These 30 articles are selected from the journals of different domains like: PLOS, ARXIV and Psychology journal of judgment and decision making (JDM), with equal number of article from each domain and are labeled by three independent annotators.

V. Experimental Setup

All the experiments are conducted in Intel core-i5 machine with 1TB HDD, 2.6GHz CPU and 8GB RAM under Windows Environment using Java [36]. The architecture used to conduct experiments are shown in Fig. 1. The architecture used to conduct experiments is shown in Fig. 1 under Windows Environment using Java [36]. The architecture used to conduct experiments is shown in Fig. 1.

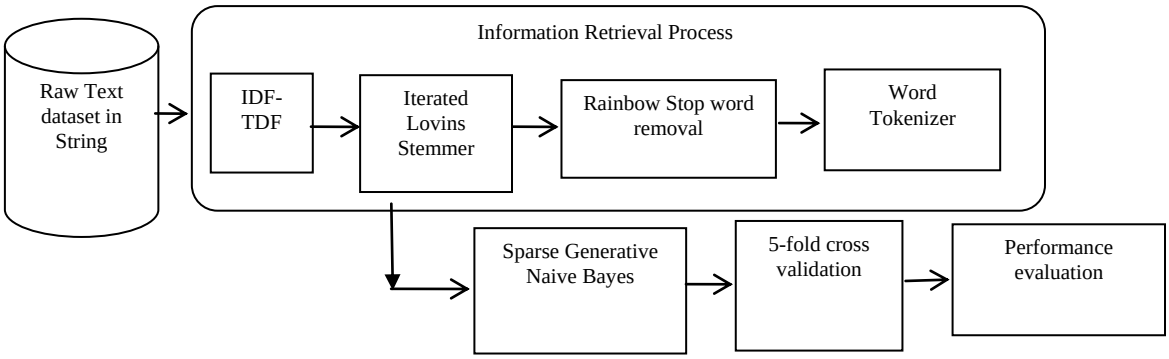

Fig.1. Proposed Methodology

Information retrieval process converts the document strings into vector as a pre-processing step. Here, the string attributes are converted to the vector attributes representing the word occurrence in the string, depending on the type of tokenizer used in the process. While, TFT transform used to set word frequencies into log $(1+f_{ij})$; IDFT transform set the word frequencies in a document into f_{ij}*log (total number of documents/ number of documents with word i), where f_{ij} denotes the frequency count of i in document j. Then, iterated lovins stemmer is used on the words, followed by removal of stop words using rainbow list. Finally, the word tokenizer is used to complete the information retrieval process.

In the next step, text classification is used with sparse generative naive Bayes classifier in a 5-fold cross validation method to obtain the efficiency and effectiveness of the proposed approach.

A summary of the datasets used is provided in Table 1.

Table 1. Description of datasets

Dataset	**features**	**instances**	**class**
Reuters C50	10913	2500	50
Review polarity	1166	2000	2
Twitter-Arabic sentiment Analysis	4363	2000	2
Sentence corpus	1356	98	4
ACLIMDB v1	1177	55733	3
20 news group	4856	19997	20

VI. Results and Discussions

In this paper, performance measures are obtained with Precision, Recall, F1-measure, model building time and classification accuracy. The main reasons of using precision-recall apart from accuracy is that one may get best accuracy for the highly skewed class distribution by always guessing the majority class.

In information retrieval system, precision, also called as positive predictive values, can be defined as the number of retrieved instances that are relevant to the query whereas recall or otherwise known as sensitivity, means the fraction of relevant instances that are retrieved.

In this scenario, there is a compromise between which one to choose. As is evident, if one envisages of entire document collection is relevant, then recall is 100% while the precision is 0%. On the other way, when one thinks of a single document is most relevant to a user query, and then the reverse is true for recall and precision value. Hence, a balance between these two is always suggested for a better model. Further, to compensate these dilemmas, F1 measure which is the harmonic mean of precision-recall may be better choice in information retrieval system. Further, win-tie- loss (w-t-l) is used to check its suitability in text mining applications, in comparison to other available methods.

Table 2 provides the experimental results obtained by our proposed approach using sparse generative Naive

Bayes (SGNB) in terms of Precision, Recall, F-score, accuracy and time taken to build the classification model.

Table 2. Experimental Results with SGNB classifier

Datasets	Avg. Precision	Avg. Recall	Avg. F-Score	Accuracy (%)	Time (sec)
Reuters C50	0.861	0.859	0.857	85.92	1.48
Review polarity-sentence token	0.831	0.831	0.831	83.1	0.72
Twitter	0.848	0.839	0.838	83.9	0.27
Sentence corpus	0.902	0.694	0.784	69.39	0.47
ACLIMDB v1	0.304	0.551	0.392	55.15	3.75
20 news group	0.797	0.867	0.792	89.95	2.14

The obtained results are further compared with Discriminative Multinomial Naive Bayes (DMNBText) text classifier in terms of % accuracy in Table 3. Then, comparison with others work is presented in Table 4.

Finally, win-tie-loss criteria are adopted to understand the effectiveness of our proposed model, which is outlined in Table 5 and Table 6.

Table 3. Comparison of % accuracy with DMNBText classifier

Datasets	Sparse generative Naive Bayes (SGNB)	DMNBText
Reuters C50	85.92	83.84
Review polarity-sentence token	83.1	80.8
twitter	83.9	69.5
Sentence corpus	69.39	94.89
ACLIMDb v1	55.15	54
20 news group	89.95	57

Table 4. Comparison of % accuracy with other baseline methods

Classifier/Dataset	ReuterC50	20 news group	IMDb	Sentence token	Twitter	Sentence corpus
SGNB (ours)	85.92	89.95	55.15	83.1	83.9	69.39
DMNBText (ours)	83.84	57	54	80.8	69.5	94.89
SVM+ Uni-gram [42]	88	-	-	-	-	-
Bisect k-means [37]	-	52.64	-	-	-	-
PGSM [37]	-	35.71	-	-	-	-
KNN [43]	-	87.57	-	-	-	-
NB [43]	-	86.71	-	-	-	-
CNN [39]	-	-	40	-	-	-
LSTM [39]	-	-	43	-	-	-
BOWSVM [40]	-	-	-	78.24	-	-
WVSVM [40]	-	-	-	78.53	-	-
BOWWVSVM [40]	-	-	-	79.67	-	-
BOW-LG [40]	-	-	-	78.24	-	-
One-hot vector CNN [40]	-	-	-	77.83	-	-
SVM [41]	-	-	-	-	68.7	-
SGD [41]	-	-	-	-	67.1	-
BNB [41]	-	-	-	-	67	-

Table 5. W-L-T for accuracy comparison-Part1

Dataset/ Classifier	SGNB (ours)	DMNBText (ours)	Bisect k-means [37]	PGSM [38]	KNN [43]	CNN [39]	LSTM [39]	NB [43]	SVM+ uni-gram [42]
20 news group	5/0/0	2/0/3	1/0/4	0/0/5	4/0/1	-	-	3/0/2	-
IMDb	3/0/0	2/0/1	-	-	-	0/0/3	1/0/2	-	-
Reuters C50	1/0/1	0/0/2	-	-	-	-	-	-	2/0/0

Table 6. W-L-T for accuracy comparison-Part2

Dataset/Classifier	SGNB (ours)	DMNBText (ours)	SVM [41]	SGD [41]	BNB [41]	BOWSVM [40]	WVSVM [40]	BOWWVSVM [40]	BOWLG [40]	One vector CNN [40]
Twitter	4/0/0	3/0/1	2/0/2	0/1/3	0/1/3	-	-	-	-	-
Sentence polarity	6/0/0	5/0/1	-	-	-	1/1/4	3/0/3	4/0/2	1/1/4	0/0/6

From the Table 5 and Table 6, it is quite evident that our proposed approach performs well in comparison to available base line methods with most number of wins in 20 newsgroup dataset, IMDb dataset (ACLIMDB v1), Twitter data and sentence polarity data except Reuters C50 with one win and one loss. In Reuters C50 dataset, SVM+Unigram method [43] is the best option with 2 wins and no loss. The proposed SGNB approach is faster, with good precision, Recall and F-score, as can be seen from Table 2 justifies its efficiency and effectiveness in variety of text classification tasks.

VII. Conclusions and Future Scope

The effectiveness of Information retrieval (IR) does not centered only at finding useful information but also largely depends on the methodology of word count through tokenization, stemming and stop word removal process as a pre-processing step before being used for text classification. Hence, these three criteria play a crucial role developing an effective and efficient IR model. The better token resulted after pre-processing is definitely less in number than the original ones, which in turn requires low memory space and finally takes less time in building the model. The proposed approach with SGNB classifier in combination of iterated Lovins stemmer, rainbow stop word removal and word tokenizer performs well in taking less time ranging from 0.27 seconds to 3.75 seconds for building the model with acceptable accuracy, precision, recall and f-score with most number of wins in comparison to all but one datasets. In future, investigation shall be carried out with some novel feature selection algorithms combining with deep learning with diverse big text dataset to understand the effectiveness of the text mining process.

References

[1] M. Rahimirad, M. Mosleh and A. M. Rahmani. Improving the Operation of Text Categorization Systems with Selecting Proper Features Based on PSO-LA, Journal of Advances in Computer Engineering and Technology, Vol. 1(2), pp. 1-8, 2015. http://jacet.srbiau.ac.ir/article_6706_6cc25826769e12ffcc494d3348913fe9.pdf

[2] Z. Elberrichi, A. Rahmoun, and Mohamed A. Bentaalah, Using WordNet for Text Categorization,The International Arab Journal of Information Technology, Vol. 5(1), pp.16-24, 2008. http://ccis2k.org/iajit/PDF/vol.5,no.1/3-37.pdf

[3] J. Chen, H. Huang, S. Tian, and Y. Qu. Feature selection for text classification with naive Bayes. Expert Systems with Applications, Vol. 36(3), pp. 5432-5435, 2009.

[4] R. Krovetz. Viewing morphology as an inference process. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 191-202, 1993. New York, NY: ACM Press.

[5] J. Huang, J. Lu and C.X. Ling. Comparing naive Bayes, decision trees, and SVM with AUC and accuracy, Third IEEE International Conference on Data Mining, 2003. ICDM 2003, DOI-10.1109/ICDM.2003.1250975

[6] C.D. Manning, P. Raghavan and H. Schu¨tze. Introduction to Information Retrieval, 1st ed. Cambridge University Press, 2008.ISBN-13: 978-0521865715

[7] M. Fernandez-Gavilanes, T. Alvarez-Lopez, J. Juncal-Martõnez, E. Costa-Montenegro, F. J. Gonzalez-Castano. Un-supervised method for Sentiment Analysis in online texts, Expert Systems With Applications, Elsevier, 2016. doi: 10.1016/j.eswa.2016.03.031

[8] W. Medhat, A. Hassan and H. Korashy. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, Vol.5, pp.1093–1113,2014. http://www.sciencedirect.com/science/article/pii/S2090447914000550.

[9] V. A. Kharde and S.S. Sonawane, Sentiment Analysis of Twitter Data: A Survey of Techniques, International Journal of Computer Applications, Vol. 139(11), pp. 5-15, April 2016. DOI: 10.5120/ijca2016908625

[10] B. Agarwal, I. Xie, O. Vovsha, R. P. Rambow. Sentiment Analysis of Twitter Data, In Proceedings of the ACL 2011Workshop on Languages in Social Media, 2011 , pp. 30-38.

[11] Po-Wei Liang, Bi-Ru Dai. Opinion Mining on Social Media Data, IEEE 14th International Conference on Mobile Data Management,Milan, Italy, June 3 - 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5, http://doi.ieeecomputersociety.org/10.1109/MDM.2013

[12] D. Davidov and A. Rappoport. Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. Coling 2010: Poster Volume, 241-249, Beijing, August 2010

[13] R. Xia, C. Zong, and S. Li. Ensemble of feature sets and classification algorithms for sentiment classification,Information Sciences: an International Journal, Vol. 181(6), pp. 1138–1152, 2011.

[14] X. Li, C. X. Ling, and H. Wang. The convergence behavior of naive Bayes on large sparse datasets. ACM Trans. Knowl. Discov. Data Vol.1(1), Article 10,(July 2016), pp.1-24. DOI: http://dx.doi.org/10.1145/2948068

[15] A. S. Altheneyan, Mohamed El Bachir Menai. Naive Bayes classifiers for authorship attribution of Arabic texts, Journal of King Saud University – Computer and Information Sciences, Vol. 26, pp. 473–484, 2014.

[16] J. K. Raulji , J. R. Saini. Stop-Word Removal Algorithm and its Implementation for Sanskrit Language, International Journal of Computer Applications, Vol. 150(2), pp. 15-17, 2016.

[17] Z. Elberrichi, A. Rahmoun, and M. A. Bentaalah. Using WordNet for Text Categorization, The International Arab Journal of Information Technology, Vol. 5(1), pp. 1-9, 2008.

[18] Mohammed A. Otair. Comparative analysis of Arabic Stemming Algorithms, International Journal of Managing Information Technology (IJMIT), Vol. 5(2), pp. 1-12, 2013.

[19] S. Lai, L. Xu, K. Liu and J. Zhao. Recurrent Convolutional Neural Networks for Text Classification, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2267-2273, 2015.

[20] J. D. Prusa, Taghi M. Khoshgoftaar, D. J. Dittman. Impact of Feature Selection Techniques for Tweet Sentiment Classification, Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, pp. 299-304, 2015, AAAI.

[21] N. Sheydaei, Mohamad Saraee and Azar Shahgholian. A novel feature selection method for text classification using association rules and clustering, Journal of Information Science, Vol. 41(1), pp. 3–15, 2015.

[22] Ferhat O¨ Zgu¨ R C Atak, Genetic Algorithm based Feature Selection in High Dimensional Text Dataset Classification, WSEAS Transactions On Information Science And Applications, Vol. 12, pp. 290-296, 2015.
[23] R. Krovetz. Viewing morphology as an inference process. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 191-202, 1993.
[24] A. Arampatzis, van der Weide, Th.P., Koster, C.H.A., and Van Bommel, P.. Linguistically-motivated Information Retrieval. Encyclopedia of Library and Information Science, published by Marcel Dekker, Inc. - New York – Basel, 2000.
[25] M. Porter. An algorithm for suffix stripping. Program, Vol. 14(3), pp. 130-137, 1980.
[26] J. B. Lovins. Development of a Stemming Algorithm, Mechanical Translation and Computational Linguistics, Vol. 11, 1968.
[27] P. Turney. Learning to Extract Keyphrases from Text, ERB-1057, National Research Council Canada, pp. 1-45, 1999.
[28] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, 1995. http://www.cs.cmu.edu/~mccallum/bow
[29] A. Puurula. Scalable text mining with sparse generative models, 2016. https://arxiv.org/pdf/1602.02332.pdf
[30] S. Jiang, Harry Zhang, Charles X. Ling, Stan Matwin: Discriminative Parameter Learning for Bayesian Networks. In: ICML 2008.
[31] UCI Machine Learning Repositiory, Reuter 50 50 Dataset. https://archive.ics.uci.edu/ml/datasets/Reuter_50_50. (Visited on 04/14/2014).
[32] Newsgroup Data Set. 2006. http://people.csail.mit.edu/20Newsgroup/
[33] N.Abdulla, N. A. Mahyoub, M. Shehab and M. Al-Ayyoub Arabic Sentiment Analysis: Corpus-based and Lexicon-based, IEEE conference on Applied Electrical Engineering and Computing Technologies (AEECT 2013), December 3-12, 2013, Amman, Jordan.
[34] C. Potts. On the negativity of negation. In Nan Li and David Lutz, eds., Proceedings of Semantics and Linguistic Theory, Vol. 20, pp. 636-659, 2011.
[35] S. Teufel and M. Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. Computational Linguistics, Vol. 28(4), pp. 409-445, 2002.
[36] I. H Witten, E Frank, MA Hall and CJ Pal. Data Mining: Practical machine learning tools and techniques, 2016. Morgan Kauffman.
[37] K. Murugesan and J. Zhang. Hybrid bisect K-means clustering algorithm. In: IEEE International Conference on Business Computing and Global Informatization (BCGIN), pp. 216–219, 2011. IEEE
[38] S. C. Tan, K. M. Ting and S. M. Teng. A general stochastic clustering method for automatic cluster discovery. Pattern Recogn. Vol. 44 (10–11), pp. 2786–2799, 2011.
[39] M. Sorostinean, K. Sana, M. Mohammad and A. Targhi. Sentiment analysis on Movie reviews, 2017.
[40] Y. Zhang and B. C. Wallace. A sensitivity analysis of (and practitioner guide to) convolutional neural network for sentiment classification, 2016. ARXIV. arXiv:1510.03820
[41] M. Nabil, M. Aly and A. F. Atiya. ASTD: Arabic Sentiments Tweets Datasets, In: Proc. Of of 2015 conference on empirical methods in Natural Language processing, pp. 2515-2519, 2015.
[42] S. Nirkhi, R. V. Dharaskar and V.M.Thakare. Authorship identification uisng generalized features and analysis of computational methods, Transaction on Machine learning and artificial Intelligence, Vol. 3(2), pp. 41-45, 2015.
[43] A. Danesh, B. Moshiri and O. Fatemi. Improve text classification accuracy based on classification fuson methods, in: Proceedings of FUSION, pp. 1-6, 2007. IEEE.

Author's Profile

Mrutyunjaya Panda holds a Ph.D degree in Computer Science from Berhampur University. He obtained his Master in Communication System Engineering from University College of Engineering, Burla, under Sambalpur University, MBA in HRM from IGNOU, New Delhi, Bachelor in Electronics and Tele-Communication Engineering from Utkal University respectively. He is having 19 years of teaching and research experience. He is presently working as Reader in the P. G. Department of Computer Science and Applications, Utkal University, Vani Vihar, Bhubaneswar, Odisha, India. He is a member of KES(Australia), IAENG(Hong Kong), ACEEE(I), IETE(I), CSI(I), ISTE(I). He has published about 70 papers in International and National journals and conferences. He has also published 5 book chapters , edited two books in Springer and authored two text books on Soft Computing Techniques and Modern approaches of Data Mining; to his credit. He is a program committee member of various international conferences. He is acting as a member of editorial board and active reviewer of various international journals. His active area of research includes Data Mining, Granular Computing, Big Data Analytics, Internet of Things, Intrusion detection and prevention. Social networking, wireless sensor networks, Image Processing, Text and Opinion Mining and Bioinformatics etc.

How to cite this paper: Mrutyunjaya Panda, " Developing an Efficient Text Pre-Processing Method with Sparse Generative Naive Bayes for Text Mining", International Journal of Modern Education and Computer Science(IJMECS), Vol.10, No.9, pp. 11-19, 2018.DOI: 10.5815/ijmecs.2018.09.02

