ISSN: 2581-3404 (Online) IF: 5.68 (SJIF)
IJIRTM, Volume-8, Issue-1, January-2024.

Enhancing Text Classification Performance through Machine Learning Techniques

Sheetal Jaiswal[1], Chetan Agrawal[2], Rashi Yadav[3]
Dept. of CSE, Radharaman Institute of Technology & Science, Bhopal, India[1, 2, 3]
sheetaljaiswal2397@gmail.com[1], chetan.agrawal12@gmail.com[2], rashi6yadav@gmail.com[3]

Abstract. *This research paper focuses on leveraging machine learning techniques to enhance text classification performance for marketing-related product reviews. The primary objective is to discern the sentiment of customer reviews, thereby aiding in improving service and product quality. Acquiring valuable online reviews presents a challenge, particularly in accurately recognizing review sentiment. The study dynamically identifies customer opinions regarding specific products, distinguishing between positive and negative sentiments. Given the significance of online reviews in purchase decision-making, this work contributes to monitoring and managing early promotion efforts. Furthermore, since reviewers often translate into actual product adopters, precise sentiment identification facilitates direct purchases. The paper presents a method utilizing Support Vector Machine (SVM) and Naive Bayes algorithms for text classification, demonstrating superior accuracy in Text Classification.*

Keywords:- Machine Learning, Text Classification, SVM, Sentiment Analysis.

Introduction

A great deal of Classification Algorithms has been introduced in previous researches in which Support Vector Machines (SVMs) has proved to be very effective and robust for Text Classification. Text Summarization is an Emerging Technique which distills the most important information from a source to produce an abridged version for a particular user .One technique of Extractive Summarization is to assign weight to each sentence as determined by some characteristics of the sentence. Then all terms in the article are re-weighted after Summarization. Terms from more important sentences will be given more weights. Some initial studies have been done to apply Summarization technique in Classification to re-weight terms the motivation of using Summarization technique in Text Classification is this features from more important sentences should be considered more valuable than other features and therefore should be given more weights in the "Bag of Words" representation. In [1] only used summarization technique to select good features and built classifier based on the reduced feature space. The result was competitive with state-of-the-art feature selection techniques. In [2] used Summarization technique to calculate the importance of sentences and re-weighted all the features. Improvement was achieved on several classifiers including Support Vector Machines. In the [3] it Employed Summarization Technique to increase the performance of Web Page Classification. [4] Investigated the effect on classification from redundant information in summaries. He reduced redundancy in summaries to different levels in order to investigate its effect on Support Vector Machines performance. Since summary can cover almost all the important information of

original document, and every reader can understand the meaning of original document by reading its summary [7], summary can be a substitute of original and user should be able to determine the category for each document by reading its summary. Based on above considerations and in order to reduce the calculation of feature selection and improve the performance of categorization, the summary instead of original document is applied to feature selection and classifier training. Moreover, a document may be assigned to more than one category, but in only researches on Hard Categorization (assigning a single category to each document) are taken into consideration. Some previous works have considered the application of Text Summarization in the task of Document Categorization. Even though these works have studied different aspects of this application, most of them have revealed, directly or indirectly, the potential of Text Summarization as a Feature Selection Technique. Some of these works have used Text Summarization (or its underlying ideas) to improve the weighting of terms and thereby the classification performance. For instance, in [5] weighted the terms by taking into account their frequency and position in the documents; whereas in [6] considered a weighting scheme that rewards the terms from the phrases selected by a Summarization Method. A more ambitious approach consists of applying Text Summarization with the aim of reducing the representation of documents and enhancing the construction of the classification model. Examples from this approach are the works in [7]. The former work is of special relevance since it showed that significant improvements can be achieved by classifying extractive summaries rather than entire documents. Finally, the work in explicitly proposes the use of summarization as a Feature Selection Technique. They applied different summarization methods based on the selection of sentences with the most important concentration of keywords or title words and compared the achieved results against those from a statistical feature selection technique, concluding that both approaches are comparable.

Machine learning based systems-Instead of relying on manually crafted rules, machine learning text classification learns to make classifications based on past observations. By using pre-labeled examples as training data, machine learning algorithms can learn the different associations between pieces of text, and that a particular output (i.e., tags) is expected for a particular input (i.e., text). A “tag” is the pre-determined classification or category that any given text could fall into. The first step towards training a machine learning NLP classifier is feature extraction: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is bag of words, where a vector represents the frequency of a word in a predefined dictionary of words. For example, if we have defined our dictionary to have the following words {This, is, the, not, awesome, bad, basketball}, and we wanted to vectorize the text “This is awesome,” we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0). Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. *sports*, *politics*) to produce a classification model: Once it’s trained with enough training samples, the machine learning model can begin to make accurate predictions. The same feature extractor is used to transform unseen text to feature sets which can be fed into the classification model to get predictions on tags (e.g. *sports*, *politics*): important application and research topic since the inception of digital documents. Today, text classification is a necessity due to the very large amount of text documents that we have to deal with daily. In general, text classification includes topic based text classification and text genre-based classification. Topic-based text categorization

classifies documents according to their topics [8]. Texts can also be written in many genres, for instance: scientific articles, news reports, movie reviews, and advertisements. Genre is defined on the way a text was created, the way it was edited, the register of language it uses, and the kind of audience to whom it is addressed. Previous work on genre classification recognized that this task differs from topic-based categorization [9]. Typically, most data for genre classification are collected from the web, through newsgroups, bulletin boards, and broadcast or printed news. They are multi-source, and consequently have different formats, different preferred vocabularies and often significantly different writing styles even for documents within one genre. Namely, the data are heterogeneous. Intuitively Text Classification is the task of classifying a document under a predefined category. More formally, if i d is a document of the entire set of documents D and {cc c 1 2,,..., n} is the set of all the categories, then text classification assigns one category j c to a document i d. As in every supervised machine learning task, an initial dataset is needed. A document may be assigned to more than one category (Ranking Classification), but in this paper only researches on Hard Categorization (assigning a single category to each document) are taken into consideration. Moreover, approaches, that take into consideration other information besides the pure text, such as hierarchical structure of the texts or date of publication, are not presented. This is because the main issue of this paper is to present techniques that exploit the most of the text of each document and perform best under this condition [10]

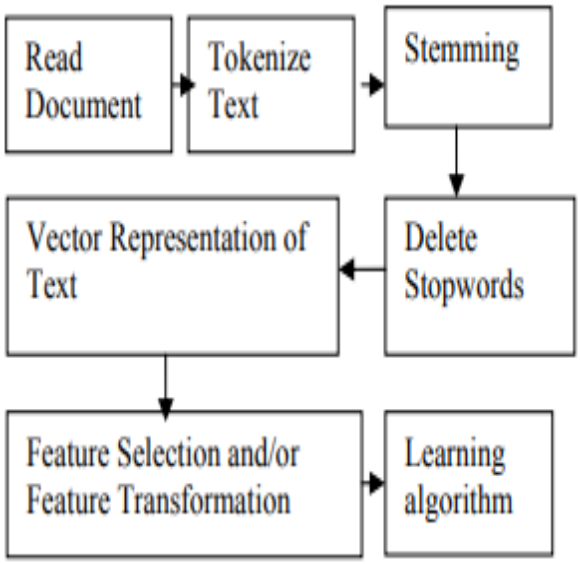

Fig. 1: Text Classification Block Diagram.

Text classification with machine learning is usually much more accurate than human-crafted rule systems, especially on complex NLP classification tasks. Also, classifiers with machine learning are easier to maintain and you can always tag new examples to learn new tasks.

Rest of the paper is organized as follow: in section II we explained various machine learning text classification algorithms, in section 3 we presents previous work done by researchers in the domain of text

classification, section 4 presents Proposed method with its algorithm and flow graph, implementation details with dataset information and its results with comparative study and discussion are given in section 5, finally we conclude our work in section 6 followed by references used.

II. Machine Learning Text Classification Algorithms

Some of the most popular machine learning algorithms for creating text classification models includes the Naive Bayes family of algorithms, support vector machines.

Naive Bayes-The Naive Bayes family of statistical algorithms are some of the most used algorithms in text classification and text analysis, overall. One of the members of that family is Multinomial Naive Bayes (MNB) with a huge advantage, that you can get really good results even when your dataset isn't very large (~ a couple of thousand tagged samples) and computational resources are scarce. Naive Bayes is based on Bayes's Theorem, which helps us compute the conditional probabilities of the occurrence of two events, based on the probabilities of the occurrence of each individual event. So we're calculating the probability of each tag for a given text, and then outputting the tag with the highest probability. The probability of A, if B is true, is equal to the probability of B, if A is true, times the probability of a being true, divided by the probability of B being true. This means that any vector that represents a text will have to contain information about the probabilities of the appearance of certain words within the texts of a given category, so that the algorithm can compute the likelihood of that text's belonging to the category.

Support Vector Machines-Support Vector Machines (SVM) is another powerful text classification machine learning algorithm, because like Naive Bayes, SVM doesn't need much training data to start providing accurate results. SVM does, however, require more computational resources than Naive Bayes, but the results are even faster and more accurate. In this we present the QP formulation for SVM classification .This is a simple representation only.

III. Related Work

Bang An et.al. [6] In recent years, Active Learning (AL) has been applied in the domain of text classification successfully. However, traditional methods need researchers to pay attention to feature extraction of datasets and different features will influence the final accuracy seriously. In this paper, we propose a new method that uses Recurrent Neutral Network (RNN) as the acquisition function in Active Learning called Deep Active Learning (DAL). For DAL, there is no need to consider how to extract features because RNN can use its internal state to process sequences of inputs. We have proved that DAL can achieve the accuracy that cannot be reached by traditional Active Learning methods when dealing with text classification. What's more, DAL can decrease the need of the great number of labelled instances for Deep Learning (DL). At the same time, we design a strategy to distribute label work to different workers. We have proved by using a proper batch size of instance, we can save much time but not decrease the model's accuracy. Based on this, we provide batch of instances for different workers and the size of batch is determined by worker's ability and scale of dataset, meanwhile, it can be updated with the performance of the workers.

Mehmet Umut SALUR et.al [7] the increase of text based content produced in electronic environment necessitated to classify these contents with advanced algorithms. For text classification, the text contents are reduced to a more structured form by being preprocessed. Many studies have examined how these pre-

processes affect the performance of classification algorithms. The main question of this study is how the Convolutional Neural Network (CNN), which is noteworthy with its classification success, is influenced by text preprocessing in the text classification. In this study, the effect of text preprocessing is examined CNN classification performance on two different data sets with Turkish content. As a result of the study, it has been determined that the text preprocessing has a weak positive effect on CNN classification performance.

Yujia Zhai et.al [8] Text classification refers to the process of automatically determining text categories based on text content in a given classification system. Text classification mainly includes several steps such as word segmentation, feature selection, weight calculation and classification performance evaluation. Among them, feature selection is a key step in text classification, which affects the classification accuracy. Feature selection can help indicate the relevance of text contents and can better classify the text. Meanwhile feature selection has a great influence on the classification result. Text classification is a very important module in text processing, and it is widely applied in areas like spam filtering, news classification, sentiment classification, and part-of-speech tagging. This paper proposes a method for extracting feature words based on Chi-square Statistics. Because the feature words that appear together or separately may differ in different situations, we classify texts by using single word and double words as features at the same time. Based on our method, we performed experiments using classical Naive Bayes and Support Vector Machine classification algorithms. The efficiency of our method was demonstrated by the comparison and analysis of experimental results.

Yunxiang Zhang et.al [9] Text classification widely exists in the fields of e-commerce and log message analysis. Besides, it is an essential module in text processing tasks. In this paper, we present a method to create an accurate and fast text classification system in both One-vs.-one and One-vs.-rest manner. Our approach, named n-BiLSTM, is used to convert natural text sentences into features similar to bag-of-words with n-gram techniques, and then the features are fed into a bidirectional LSTM. The two components are able to take better advantages of multi-scale feature representation and context information. Finally, the whole system is evaluated using two labelled movie review datasets, IMDB and SSTb, to test one-vs.-one and one-vs.-rest performances respectively. The results obtained show that our n-BiLSTM algorithm is superior to the basic LSTM and bidirectional LSTM algorithms.

Tengjun Yao et.al [10] most text classification models based on traditional machine learning algorithms has problems such as curse of dimensionality and poor performance. In order to solve the above problems, this paper proposes a text classification model based on fast Text. Our model explores the important information contained in the text through the feature engineering, and obtains the low-dimensional, continuous and high-quality text representation through the fast Text algorithm. The experiment is based on Python to classify the text dataset of “user comment data emotional polarity judgment” in Baidu Dianshi platform. In the emotional polarity judgment task, the experimental results show that the precision, recall and F values of our model are superior to the model based on traditional machine learning algorithms and have excellent classification performance.

Yajing Ma et.al [11] Text classification technology refers to the process of supervised learning of specified texts based on fixed rules. This paper describes the development process of text classification and its method techniques, including text representation, feature selection and classification algorithms, and draws the basic ideas, advantages and disadvantages of several current mainstream classification techniques

Problem Statement - The existing causing the ineffectiveness for text classification. Clustering users by reviews is more challenging than in the case of long documents associated with them as it is difficult to track users' reviews in streaming sparse data. Loss of information during the clustering. This may lead to the incorrect clustering result. Automatic text classification plays an important role in many applications. These applications can be sorted in a timely, correct and correct manner, directly, classifying and delivering appropriate documents. It is the basic building block of a wide range of contexts from document indexing to document filtering, verbatim ambiguity, hierarchical filling of web catalog and all applications that generally require document organization or selective and adaptive document submission. The term "text classification" is also called "monitored text classification". It has the labels on the training dataset in advance and automatically assigns documents to a predefined set of categories. With the improvement of the availability of a large amount of textual information on the Internet, word processing technology has attracted more and more interest. An increasingly important specific text extraction problem involves the detection of textual expressions that refer to views on specific topics and services. The second text extraction problem has also attracted increasing interest, which is to determine the geographical location that is most relevant to the content of a particular document.

IV. Proposed Work

The proposed model is introduced to overcome all the disadvantages that arise in the existing system. It will effectively done text classify on online reviews that much helpful for the reviewers to easily identify opinion about the particular product. Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness. However, its performance is often degraded because it does not model text well. Schneider addressed the problems and show that they can be solved by some simple corrections.

Klopotek and Woch presented results of empirical evaluation of a Bayesian multinet classifier based on a new method of learning very large tree-like Bayesian networks .The study suggests that tree-like Bayesian networks are able to handle a text classification task in one hundred thousand variables with sufficient speed and accuracy. Support vector machines (SVM), when applied to text classification provide excellent precision, but poor recall. One means of customizing SVMs to improve recall, is to adjust the threshold associated with an SVM. Shanahan and Roma described an automatic process for adjusting the thresholds of generic SVM with better results. Johnson et al. described a fast decision tree construction algorithm that takes advantage of the sparsity of text data, and a rule simplification method that converts the decision tree into a logically equivalent rule set Lim proposed a method which improves performance of kNN based text classification by using well estimated parameters .Some variants of the kNN method with different decision functions, k values, and feature sets were proposed and evaluated to find out adequate parameters

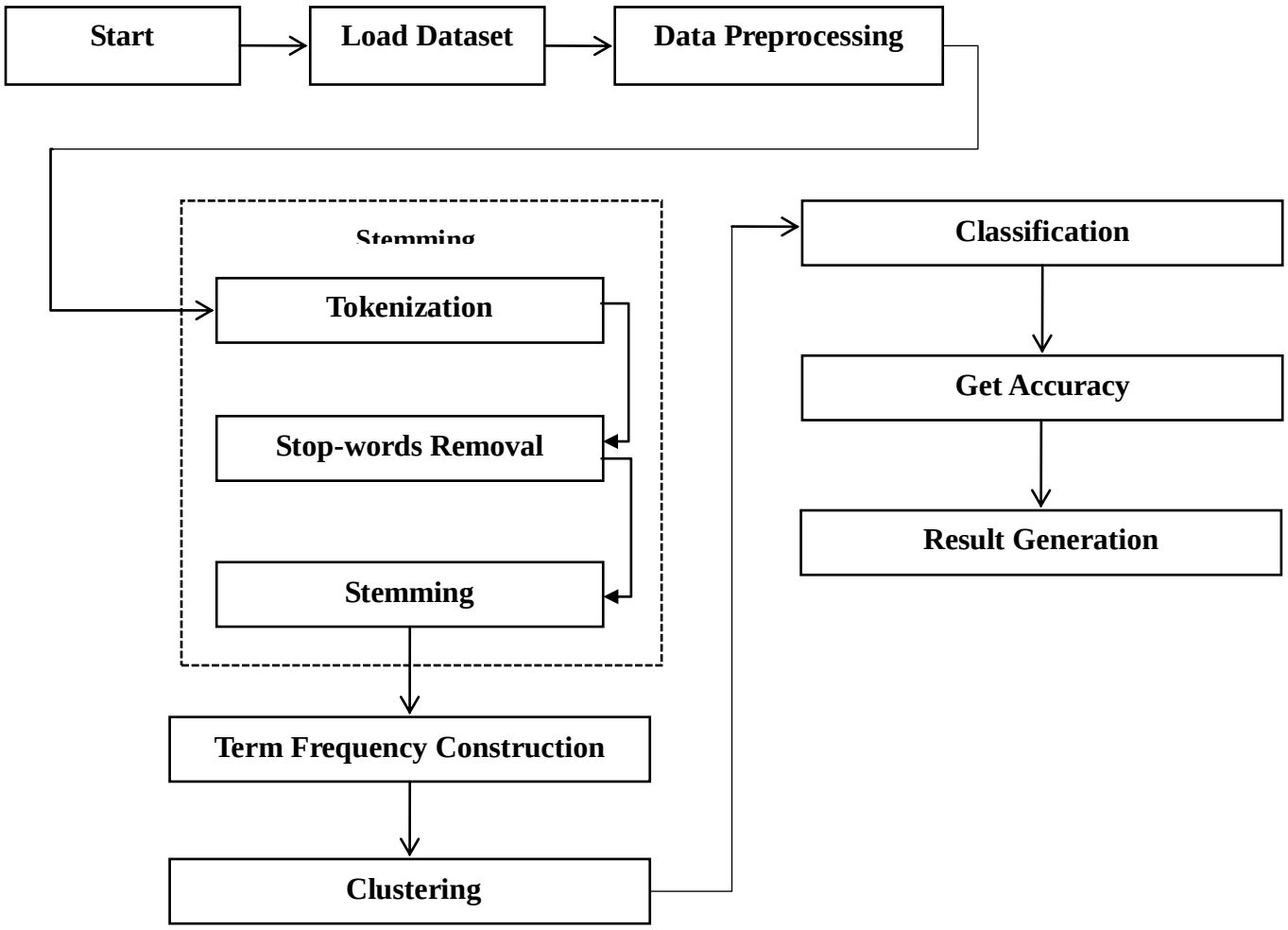

Fig. 2: Flow Diagram

Proposed Algorithm:

- Step 1: Data Selection and Loading

dataset = load_dataset("product_reviews.csv")

- Step 2: Data Preprocessing

cleaned_data = preprocess_data(dataset)

- Step 3: Tokenization

tokenized_data = tokenize_data(cleaned_data)

- Step 4: Stopwords Removal

stopwords_removed_data = remove_stopwords(tokenized_data)

- Step 5: Stemming

stemmed_data = apply_stemming(stopwords_removed_data)

- Step 6: Term Frequency Construction

term_frequency_matrix = construct_term_frequency_matrix(stemmed_data)

- Step 7: Clustering

clusters = perform_clustering(term_frequency_matrix)

- Step 8: Classification of Reviews

svm_classifier = train_svm_classifier(term_frequency_matrix, labels)

- Step 9: Result Generation

evaluation_metrics = evaluate_classifier(svm_classifier, test_data, test_labels)

- generate_results(evaluation_metrics)

Data Selection And Loading- Data selection is the process of selecting the appropriate data set for processing.The dataset which contains the fields of user name, product id, product name, brand name, category, rating and the reviews. The selected dataset is going to be used for identifying the customer opinion about the product. The dynamically distributed reviews are detected from the Amazon product reviews.

Data Preprocessing- The Data preprocessing is the process of detecting, correcting or removing, corrupt or inaccurate records from the dataset. The records which provide the incorrect clustering results are detected and removed from the dataset. In this process, we are going to delete the records that contain empty fields.

Tokenization- Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences.

Stopwords Removal- Stop words are natural language words which have very little meaning, such as "and", "the", "a", "an", and similar words. The stop words are detected from the reviews and it's removed.

Stemming- Stemming is the process of converting the words of a sentence to its non-changing portions. The porter stemming algorithm is used for stemming the words.

Term Frequency construction: After the stemming process, the term frequencies are constructed from the stemming words.

Term Frequency Construction- Term frequency (TF) is used in connection with information retrieval and shows how frequently an expression (term, word) occurs in a document.Term frequency indicates the significance of a particular term within the overall document. This value is often mentioned in the context of inverse document frequency IDF.

Clustering- It's the process of dynamically clustering the users based on their reviews. The K-means clustering algorithm is used for dynamically clustering the users based on the positive and negative opinion.

Prediction- Classification is used to classify each item in a set of data into one of predefined set of classes or groups. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

V. Result and Discussion

Recently in the area of Machine Learning the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers. Numerous methods have been suggested for the creation of ensemble of classifiers. Mechanisms that are used to build ensemble of classifiers include: i) Using different subset of training data with a single learning method, ii) Using different training parameters with a single training method (e.g. using different initial weights for each neural network in an ensemble), iii) Using different learning methods. In the context of combining multiple classifiers for text categorization, a number of researchers have shown that combining different classifiers can improve classification accuracy.

Dataset: The "Product_Reviews" dataset is a structured dataset containing product reviews collected from an online retail platform. It comprises text data and sentiment labels, making it suitable for sentiment analysis and text classification tasks. Below is a detailed description of the dataset:

Attributes:

- Review Text (Textual Data): This column contains the text of product reviews written by customers. Each review is a textual representation of the customer's opinion about a specific product.
- Sentiment Label (Target Variable): The sentiment label associated with each review indicates whether the review expresses a positive or negative sentiment towards the product. It serves as the target variable for sentiment analysis and classification tasks.

Dataset Characteristics:

- Size: The dataset contains a total of 10,000 product reviews, with each review accompanied by its corresponding sentiment label.
- Balance: The dataset is balanced in terms of sentiment distribution, with an equal number of positive and negative reviews. This balance ensures that the classification model is trained on an equal representation of both classes.
- Diversity: The reviews cover a diverse range of products from various categories, including electronics, clothing, home appliances, books, and more. This diversity allows for generalization across different product domains.
- Text Variation: The reviews exhibit variation in terms of language, writing style, and sentiment expression. Some reviews may be concise and straightforward, while others may contain detailed descriptions or emotional language.

PERFORMANCE ESTIMATION

Process performance is measured based on performance indicators such as precision, sensitivity, specificity, or time consumption.

TP- is total number of properly categorised prospects (true positives).

TN- is total number of poorly classified prospects (true negative numbers).

FN- is total number of false rejections, which represents the number of false pixels of foreground pixels classified as background (false negatives).

FP- is total number of false positives, which means that pixels are mistakenly classified as foreground (false positives). Calculate presentation value for each frame of input video based on overhead indicators.

Accuracy: Precision is an indicator for evaluating classification models. Informally, precision is part of the correct prediction of our model. Formally, precision has the following definition:

Accuracy = correct number of predictions, total number of predictions
For binary classification, precision can also be calculated according to positive and negative, as follows:

Accuracy = TP + TNTP + TN + FP + FN

Where TP = true positive, TN = true negative, FP = false positive, FN = false negative

Table 1 Comparison with existing work

	Classifier	Accuracy	Precision	Recall	F-Measure
Previous work	RNN	75	78	69	74
Proposed work	Naïve Byes	79.99	80	75.4	71.5
	SVM	89.99	90.2	88.6	84.2

VI. Conclusion

This work enhances the performance of the overall clustering and prediction. It finds the different clusters and its summary effectively and quickly. The sparsity problem and reduces the information loss. The accuracy of the clustering result is highly increased. The text classification problem is an Artificial Intelligence research topic, especially given the vast number of documents available in the form of web pages and other electronic texts like emails, discussion forum postings and other electronic documents. It has observed that even for a specified classification method, classification performances of the classifiers based on different training text corpuses are different; and in some cases such differences are quite substantial. This observation implies that a) classifier performance is relevant to its training corpus in some degree, and b) good or high quality training corpuses may derive classifiers of good performance. Unfortunately, up to now little research work in the literature has been seen on how to exploit training text corpuses to improve classifier's performance. This research can be extended to include Multi-Lingual news articles for text classification by using the proposed approach in this work. The dataset used in the present work can be extended to larger dataset for the more confident results.

References

1. Z. Toh and J. Su, "Improving aspect based sentiment analysis using neural network features,," 2016.
2. P. Xie, Y. Pei, Y. Xie, and E. Xing, "Mining user interests from personal photos," 2015.
3. W. Chen, J. Wang, Y. Zhang, H. Yan, and X. Li, "User based aggregation for biterm topic model," 2015.
4. S. Liang, Z. Ren, Y. Zhao, J. Ma, E. Yilmaz, and M. D. Rijke, "Inferring dynamic user interests in streams of short texts for user clustering", 2017.
5. T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda, "Online multiscale dynamic topic models," 2010.
6. BeiHang University Beijing, China No.37 XueYuan Road (0086)18612210095 anbang@buaa.edu.cn Wenjun Wu BeiHang University Beijing, Deep Active Learning for Text

Classification Bang An China No.37 XueYuan Road (0086)13641084384 ICVISP 2018, August 27–29, 2018, Las Vegas, NV, USA © 2018 Association for Computing
7. Machinery. Mehmet Umut SALUR;İlhan AYDIN The Impact of Preprocessing on Classification Performance in Convolutional Neural Networks for Turkish Text 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) Year: 2018 DOI: 10.1109/ IEEE Malatya, Turkey, Turkey
8. Yujia Zhai;Wei Song;Xianjun Liu;Lizhen Liu;Xinlei Zhao A Chi-Square Statistics Based Feature Selection Method in Text Classification 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) Year: 2018 DOI: 10.1109/ IEEE Beijing, China, China
9. Yunxiang Zhang;Zhuyi Rao n-BiLSTM: BiLSTM with n-gram Features for Text Classification 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC) Year: 2020 DOI: 10.1109/ IEEE Chongqing, China, China
10. Tengjun Yao;Zhengang Zhai;Bingtao Gao Text Classification Model Based on fastText 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS) Year: 2020 DOI: 10.1109/ IEEE Dalian, China, China
11. Yajing Ma;Yonghong Li;Xiaolong Wu;Xiang Zhang Chinese Text Classification Review 2018 9th International Conference on Information Technology in Medicine and Education (ITME) Year: 2018 DOI: 10.1109/ IEEE Hangzhou, China
12. .K. Dave, S. Lawrence, and D. M. Pennock, ``Mining the peanut gallery: Opinion extraction and semantic classication of product reviews," in Proc. WWW, Budapest, Hungary, 2003.
13. C. Whitelaw, N. Garg, and S. Argamon, ``Using appraisal groups for sentiment analysis," in Proc. CIKM, Bremen, Germany, 2005.
14. S. R. Das and M. Y. Chen, ``Yahoo! For Amazon: Sentiment extraction from small talk on the Web,"2007.
15. Devitt and K. Ahmad, ``Sentiment polarity identication innancial news: A cohesion-based approach," in Proc. ACL, Prague, Czech Republic, 2007, pp. 984_991.
16. Liu, ``Sentiment analysis and opinion mining," Synth. Lectures Human Lang. Technol., vol. 5, no. 1, pp. 1_167, 2012.
17. V. Hatzivassiloglou and J. M. Wiebe, ``Effects of adjective orientation and gradability on sentence subjectivity," in Proc. COLING, Saarbrücken, Germany, 2000.
18. J. M. Wiebe, ``Learning subjective adjectives from corpora," in Proc. AAAI/IAAI, Menlo Park, CA, USA, 2000.
19. J.Wiebe, T.Wilson, and M. Bell, ``Identifying collocations for recognizing opinions," in Proc. ACL/EACL, Toulouse, France, 2001.
20. P. D. Turney, ``Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classi_cation of reviews," in Proc. ACL, Philadelphia, PA, USA, 2002, pp. 417_424.