International Journal of Research in Advent Technology (IJRAT) Special Issue
E-ISSN: 2321-9637
Available online at www.ijrat.org
*International Conference "*INTELINC 18*", 12th & 13th October 2018*

Preprocessing: The first step towards Text Document Classification - A Review

N. Rajkumar[1], T.S. Subashini[2], K. Rajan[3], V. Ramalingam[4]
[1, 2, 4]*Department of Computer Science and Engineering, Annamalai University*
[1]*raju.prg@gmail.com*, [2]*rtramsuba@gmail.com*, [4]*aucsevr@gmail.com*
[3]*Department of Computer Engineering, Muthiah Polytechnic College, Annamalainagar*
[3]*rajankperumal@yahoo.co.in*

Abstract- As text data in digital format both in online and offline mode is increasing exponentially nowadays, managing and retrieving these documents becomes difficult. Many natural language processing (NLP) such as archival, retrieval, query response, information summarization, etc., highly rely on the efficient categorization of text documents. This has induced researchers to apply machine learning algorithms to automatically categorize documents based on languages and within documents belonging to the same language to devise methods which classifies documents based on its context. 'Preprocessing', is the first and foremost as well as an important stage in text categorization process which can take more than 50% time or effort of the entire classification process [1]. Thus the importance of preprocessing has made us to carry out this study which reviews the state-of –the-art elements of preprocessing for text classification. This will help other researchers to have a bird's eye view of the research so far being carried out in this domain and to assist them to propose new methods for text classification.
Keywords: Natural language processing (NLP), feature extraction, pre-processing, text classification

1. INTRODUCTION

Nowadays, as online as well as offline text data is increasing drastically, the need to extract only the required information from this huge volume of data is increasing rapidly this has given rise to a new field namely Text Mining which analyses natural language text to extract useful information needed for a specific purpose. In comparison to numerical data, text is unstructured, ambiguous, and difficult to process. Text mining includes text summarization, text classification and text clustering.

The goal of text classification is to assign some piece of text to one or more predefined classes or categories.

Text classification algorithms try to classify documents into a set of pre-defined document class based on the content contained in that document. The content could be text, tweets, email, customer feedback, emotions, product review etc,. Text classification has a wide range of applications categorizing news article into various topics such as sports, politics, cinema, business etc., reviews for predicting customer sentiments, filtering spam mails, analyzing the customer queries behavior and intent. This paper focuses on text classification work which falls under two categories namely

i. Works where the classification task is applied only in a particular domain such as industry, finance, and medicine, Internet, (analysis of email, web pages, websites, and tweets) etc.
ii. Works on text classification with a specific purpose. For example, tweet language classification, classification of emotions etc.

Many machine learning algorithms related to text classification use a statistical model, where decisions are made following a probabilistic approach. More recently deep learning approaches were also applied with extremely good results. The input data is composed of text fragments that can be viewed as simple sequences of words, full phrases or even full documents. The text is transformed and some weights are assigned to different features of the data. The machine learning models are trained on the input data and can later be used when given new, unfamiliar input.

2. STEPS IN TEXT CLASSIFICATION PROCESS

The various steps involved in text classification are illustrated in Fig 1. By applying pattern classification algorithms The text classification process tries to

classify a set of given documents into different categories.

Corpus creation is the primary step where documents belonging to different categories and formats are collected which will be used during feature extraction phase and during training and testing the classifier. Existence of elaborate text corpora will help to improve the results of the computer vision methods applied for the text classification tasks. Earlier printed text namely books, magazines and newspaper served as the main source for the creation of Corpus. But with the unprecedented growth of Internet today, the World Wide Web with a huge volume of information available, has become an indispensible resource for corpus creation. As creating one's own corpus dataset is a very time consuming task in most cases, freely available online datasets are widely used by researchers. Online corpora are available for English language, European languages, Middle Eastern language and East Asian languages. Sometimes Corpus is tailor made in accordance with the domain under investigation.

Input Validation ensures that the document contains onlytext belonging to the language of study and words or sentences which are not valid are simply removed before further processing. The words preset in the corpus and the features created play a vital role and improving the performance of a text classification system. Noisy corpus refers to unimportant entities of the text such as punctuation marks, text of other languages, numbers, hypertext and hyperlinks, etc. Removing these noises from the document

would increase the accuracy, as the dimension of the feature set decreases. However care should be taken to remove these entities only in cases they are not deemed to be of any significance. Certain applications such as sentiment analysis, emoticons such as :), :P, are important and as such they should not be eliminated.

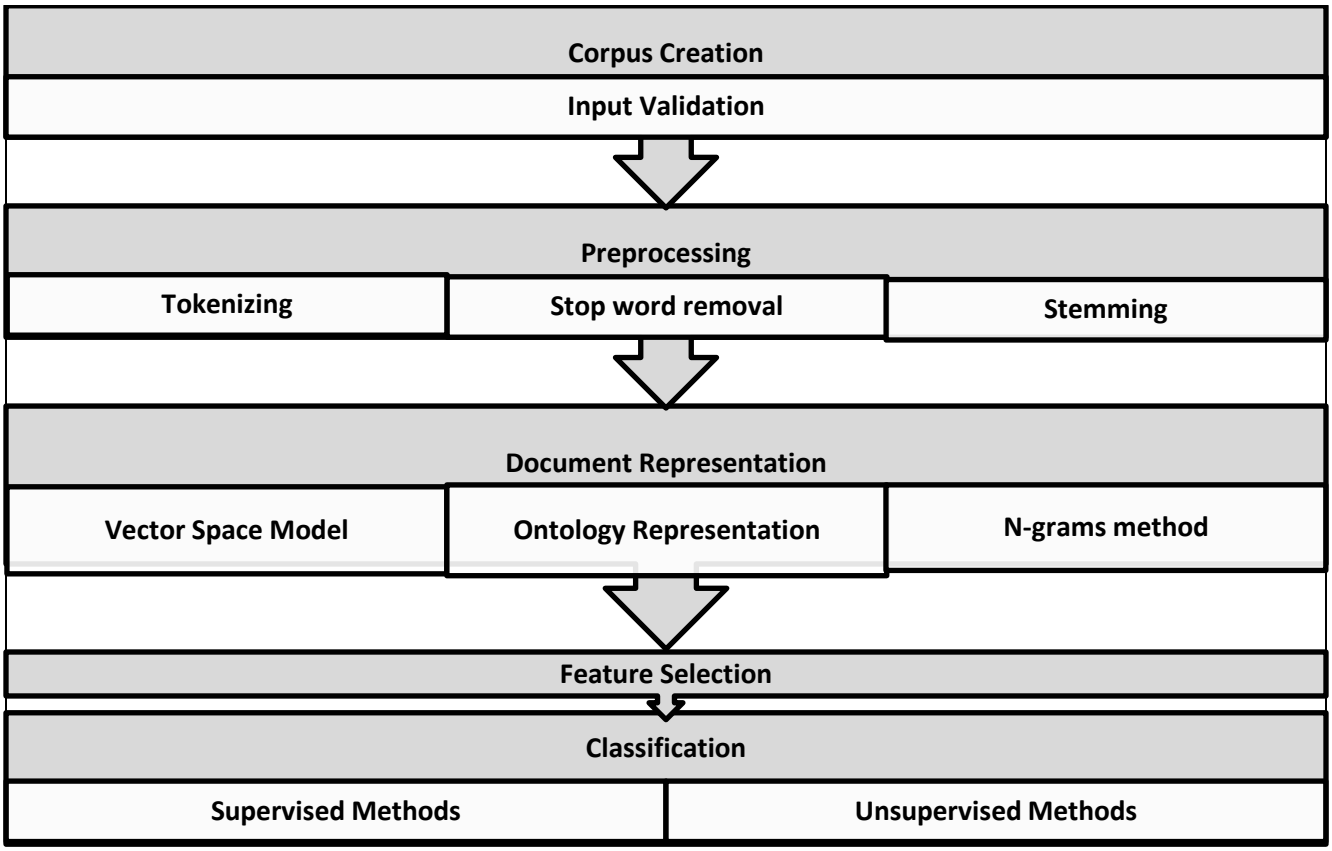

Fig. 1 Text Classification Framework

Pre-processing

A document may contain text in various forms such as words, sentences, tweets, multiple paragraphs, special characters; emoticons etc. Preprocessing step tries to make this text more meaningful for the classification algorithm to process it in an efficient way. In other words the goal behind preprocessing is to represent each document as a feature vector, that is, to separate the text into individual words.

The preprocessing **phase** includes various sub-phases namely Tokenization **Tokenization** divides the documents into tokens which is a sequence of characters, which excludes white spaces and punctuation. **Stop words** are irrelevant words though frequently occurring and which play no significant role in classifying the document. So stop words are eliminated before further processing. The next step is **stemming** which reduces the words to its stem by removing suffixes and prefixes if any. Finally the output of stemming is fed to the **Morphological Analyser,** which produces the root word by analyzing whether the given stem is inflected or not. **Section 3** gives an elaborate account on the various methods employed by researchers for preprocessing text for document classification.

The next phase is **Document representation** which transforms the whole document into a vector of words. Vector space model [16] is the document representation widely seen in the literature where the collection of *n* documents is represented as an *m x n* array where each document is a vector of dimension *m*. However, because of its huge dimension and due to lack of correlation with adjacent words, this vector cannot be straight away fed to the classifier.

Next **Feature Selection** helps to build a word vector of smaller dimension by selecting only a subset of terms/words from the original document. Based on some feature evaluation metrics such as information gain (IG), and term weighing schemes such binary term frequency and simple term frequency which evaluates the number of occurrences of a word in the document etc, A threshold is fixed and if the number of occurrence of the word is above this threshold it is retained and the remaining words are discarded.

In yet another representation, each document vector is associated with weights reflecting the number of occurrences of the terms in the document multiplied by the inverse of their frequency *(tf x idf).* This ensures that, frequently occurring words in a document vector though rarely occurring in the entire collection are of great significance in classifying documents due to its high discriminative power.

The next step is text **classification.** Machine learning algorithms could be applied for document classification. The word vector obtained after the feature selection phase could be fed into any of the supervised or unsupervised classifiers such as K-Nearest Neighbor, K-means, Neural Networks, Support Vector machine etc., to classify the given document into predefined document category.

This paper aims to present a survey of the work done so far under preprocessing of text for classification. Though some survey papers could be seen in the literature which is still useful to the research community, today automatic text classification is needed for a wide variety of government, research and business needs. This warrants survey of latest studies and systematic overview of all the **allied procedures/ preprocessing techniques** other than the machine learning techniques which have been better addressed in the earlier surveys.

3. A REVIEW ON PREPROCESSING FOR TEXT CLASSIFICATION

Out of various steps involved, this section reviews literature related to pre-processing stage alone. The text-features that are more representative of the different text categories considered are obtained in this stage. This phase is the most critical and complex process that leads to the representation of each document by a select set of features. The corpus which is a collection of unstructured data is taken as input for this stage. First the text document is split into tokens using space as the delimiter. Then stop-words which are of little or no significance are eliminated and then stemmer algorithm or morphological analyzer is employed that recognizes "stem" words embedded in other words.

3.1 Tokenization

A document is considered as a string, and using space as the delimiter the document is partioned into a set of words or tokens. In this process some special characters such as punctuation, full stop etc are discarded. The tokenization process is language-dependent. Below is shown a Tamil text which has been split into words.

இந்தியாவில் சேமிக்கும் பழக்கம் அதிகம். இந்நிலையில் மக்களின் சேமிப்புப் பணத்தை பாதுகாக்க வேண்டிய பொறுப்பும் கடமையும் அரசுக்கு உள்ளது.

The example Tamil Text is split into tokens as given below.

'இந்தியாவில்', 'சேமிக்கும்', 'பழக்கம்', 'அதிகம்', 'இந்நிலையில்', 'மக்களின்', 'சேமிப்புப்', 'பணத்தை', 'பாதுகாக்க', 'வேண்டிய', 'பொறுப்பும்', 'கடமையும்', 'அரசுக்கு', 'உள்ளது'

3.2 Stop Word Removal

Stop-word removal is a kind of data cleaning which tries to eliminate frequently occurring words which are considered to have no relevance or significance in the text classification process. These words simply act like synapses which aid in sentence formation. Conjunctions, prepositions, adverbs, articles of English language are examples of stop-words.

As such, stop-word removal is the important and primary step before documents are processed and before making queries. Stop-word removal increases system performance and speeds processing of text.

For example, the stop words removed from a Tamil document given below is shown in bold face.

எம்ஜிஆர் ஆட்சி தமிழகத்தின் பொற்காலமாக இருந்தது **என**, வேல்ஸ் பல்கலை.யில் நடந்த எம்ஜிஆர் நூற்றாண்டு விழாவில், ஆளுநர் பன்வாரிலால் புரோஹித் பேசினார். பல்லாவரம் வேல்ஸ் பல்கலை.யில், எம்ஜிஆர் நூற்றாண்டு விழாவையொட்டி, உலக எம்ஜிஆர் பேரவை பிரதிநிதிகள் மாநாடு நேற்று நடைபெற்றது. **இதில்**, தமிழக ஆளுநர் பன்வாரிலால் புரோஹித், எம்ஜிஆர் பல்கலை. ஆராய்ச்சி மைய வேந்தர் ஏ.சி. சண்முகம், சென்னை மாநகராட்சியின் முன்னாள் மேயர் சைதை துரைசாமி, அப்துல்கலாமின் முன்னாள் அறிவியல் ஆலோசகர் பொன்ராஜ், வேல்ஸ் பல்கலை. வேந்தர் ஐசரி கணேஷ், கவிஞர் வைரமுத்து, இலங்கை கல்வி அமைச்சர் ராதாகிருஷ்ணன் **மற்றும்** பலர் கலந்து கொண்டனர்.

The stop words detected are **என, இதில், மற்றும்**

In [2] several retrieval techniques and their potential in improving Arabic retrieval based on a finite state machine. effectiveness were explored. Three stop lists viz, general stop list, corpus based stop-list and combined stop list were used to carry out the comparative study. The impact of term weighting and stop words on Arabic retrieval were compared. In [6] stop-words in Arabic language have been removed by a method. By using frequency list authors in [3] arrived at a list of stop-words from a Gujarati corpus.

A stop-word removal algorithm based on dictionary based approach and its implementation for Sanskrit language is explored in [4].

In [5], a stop-word removal algorithm for Hindi documents is proposed which is based on DFA implementation. For DFA implementation, it uses json objects which take care of current state, accepting state, start state, final state, transitions and input characters. 200 Hindi documents consisting of movie review data collected from the internet were taken as input and the proposed system returns the documents with stop-words removed from them.

Eleven rules were proposed to exhaustively identify Gujarati stop words in [7]. The proposed algorithm was tested on nearly 600 Gujarati documents and results show that the proposed approach is successful in categorizing Gujarati documents.

A new method, called term-based random sampling was introduced by the authors in [8] which is based on the Kullback-Leibler divergence measure. This measure tells us how informative a term is and accordingly a term may be considered to be a stop-word or not.

Inter and intra cluster similarity approach has been proposed in [9] which try to find words which when removed increases the dissimilarity between documents in different clusters, and decreases dissimilarity between documents within the same cluster.

The authors in [10] developed a Persian information retrieval system for which they proposed an aggregated method to eliminate the Persian stop-words. Using part of speech tagging and analyzing statistical features of terms, the proposed method tries to enhance the accuracy of retrieval and minimize potential side effects of removing informative terms. The experiment results depicted good average precision, and better overall response time.

A general and domain specific stop-word list was built by authors in [11] using a new method where the information content of a word and its prediction capacity is estimated. A word could be a candidate stop-word provided it does not have a minimum information content and prediction capacity.

A stop word extraction approach for Chinese language based on statistical and information models

is proposed in [12]. Based on the probability distribution, the statistical model derives the stop words and the information model measures the significance of a word using information theory. Results from these two models are aggregated to generate the stop word list.

3.3 Stemmimg

Stemming programs are also referred to as stemming algorithms or stemmers. The purpose of stemmers is to produce morphological variants of a root or base word. For example the words "runs", "ran", "running", are reduced to the root word "run.

Though earlier, stemmers were developed for English language alone [13], the corpus growth of other languages facilitated by Unicode resulted in the research community to turn their attention towards developing stemmers for other languages too. The basic idea of some stemming algorithms along with their advantages and limitations are tabulated in Table 1.

In Indian languages, the initial work reported was on building a Hindi stemmer [14], [15]. In Kannada morphology analyzer and generator using tree architecture was reported in [16]. A Bengali stemmer for spell checking Bengali language was reported in [17].

The stemming work proposed in [18] uses a statistical approach which uses String similarity Distance (SD) to stem words of Kokborok language. The authors calculated the SD for all the related words, and then found the median of the distance for all the words and the highest occurring SD from the result. Over stemming is observed in some cases. And the reason for over stemming has been observed due to frequencies of words occurrences. Clustering technique has been employed in [19], to stem Bangla words. The proposed stemmer analyzes the prefixes and suffixes of all the word forms present in the document. Then the words having the same root are formed into a cluster with the root word as the cluster centroid.

Table 1: Stemming Methods; An overview

Algorithm	principle	Example	Advantage	Limitation
Porter's Stemmer algorithm	Removes suffixes to form root word	Removal of various suffixes such as ED, ING, ION, IONS from the input word. Example: RELATED, RELATIONS, RELATING, RELATIONS. The removal od suffix from all these word gives the root word RELATE.	It produces the best output as compared to other stemmers and it has less error rate.	Morphological variants produced are not always real words
Lovins Stemmer	Makes use of look up table. The longest suffix from a word id first removed and then recoded to convert this stem into valid words.	Dropping-> dropp ->drop	Handles irregular plurals like mice' and 'mouse' etc . and double letter word such as dropping' which has been stemmed to 'drop'	Takes more time. Sometimes fail to stem correctly as not all suffixes are present.
Dawson Stemmer	It uses more number of suffixes thn Lovins and suffixes are stored in the reversed order indexed by their length and last letter		more suffixes used than Lovins and it is fast	Too complex to implement.
Krovetz Stemmer	First the plural form of a word is converted to its	'children' -> 'child'	It is light in nature and can	If the document size is large , it

	singular form and then the past tense of a word is converted to its present tense and the suffix 'ing' is removed		be used as pre-stemmer for other stemmers	does not perform very effectively.
N-Gram Stemmer	An n-gram is a set of n consecutive characters extracted from a word in which similar words will have a high proportion of n-grams in common.	Program -> 'pr', 'ro', 'og', 'gr', 'ra, 'am'	It is language independent and hence very useful in many applications.	requires a significant amount of memory and storage for creating and storing the ngrams and indexes

An exhaustive study on the stemming methods was carried out in [20] and the study revealed that the statistical stemmer performance is far superior than rule-based stemmers though time consuming. Rule dependent stemmer like porter stemmer is good choice for English document processing but it is language dependent. The study on the various stemming methods carried out in [21], revealed that each stemmer is better in one aspect and scores better than others while lacks in some other aspect. There is still scope to design a perfect stemmer that satisfies all the requirements.

A template based stemmer for Urdu language is proposed in [22] which removes the affixes and infixes from the input text. The results of the study prove that maximum matching affix approach could be effectively employed to develop a stemmer for Urdu language. The authors in [23] presented a comparison of various stemming methods and concluded that decision based approach is mainly applicable for a specific language whereas statistical approach is language independent. The authors in [24] studied about various stemming algorithms and its efficiency on various Indian and Non-Indian languages.

The stemmer based on Brute Force algorithm was developed for Punjabi language [25] and a stemmer based on Finite State Automata (FSA) method to stem the Malayalam words is reported in [26].

4. CONCLUSION

In the literature number of works on stop word removal could be seen as elimination of stop words would definitely lead to effective indexing in all Natural Language Applications. It is almost useful in most cases to apply some form of stemming, as it leads to better document summarization as well as retrieval. Dictionary based approach, Clustering and term frequency are the widely used methodologies to remove stop words in all works irrespective of the language under study. The survey on stemming shows that there only a few works and the stemmers proposed are language specific and the problem of finding the optimal algorithm as well finding the best parameters still remains an open problem for researchers to explore.

REFERENCES

[1] Katharina, M. and Martin, S., "The Mining Mart Approach to Knowledge Discovery in Databases"

[2] , Intelligent Technologies for Information Analysis, Springer, Pp. 47-65.

[3] A. El-Khair, "Effects of stop words elimination for arabic information retrieval: a comparative study", International Journal of Computing & Information Sciences, vol. 4, no. 3, pp. 119-133, 2006.

[4] Ashish T, Kothari M and Pinkesh P, "Pre-Processing Phase of Text Summarization Based on Gujarati Language", International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Vol-2, Iss-4, July 2014.

[5] Jaideepsinh K. Raulji , Jatinderkumar R. Saini, "Stop-Word Removal Algorithm and its Implementation for Sanskrit Language", International Journal of Computer Applications (0975 – 8887) Volume 150 – No.2, 2016

[6] V. Jha, N. Manjunath, P. D. Shenoy and K. R. Venugopal, "HSRA: Hindi stopword removal algorithm," 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), Durgapur, 2016, pp. 1-5.

[7] R. Al-Shalabi, G. Kanaan, J. M. Jaam, A. Hasnah, E. Hilat, "Stop-word removal algorithm

for arabic language", Proceedings of 1st International Conference on Information & Communication Technologies: from Theory to Applications CTTA '04, pp. 545-550, 2004.

[8] Rakholia R.M., Saini J.R. "A Rule-Based Approach to Identify Stop Words for Gujarati Language", Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications. Advances in Intelligent Systems and Computing, vol 515. Springer, Singapore, 2017

[9] Tsz-Wai Lo, Rachel & He, Ben & Ounis, Iadh. (2005). Automatically Building a Stopword List for an Information Retrieval System.. JDIM. 3. 3-8.

[10] Popova S., Krivosheeva T., Korenevsky M.," Automatic Stop List Generation for Clustering Recognition Results of Call Center Recordings". Speech and Computer. SPECOM 2014. Lecture Notes in Computer Science, vol 8773. Springer, Cham, 2014

[11] M. Yaghoub-Zadeh-Fard, B. Minaei-Bidgoli, S. Rahmani and S. Shahrivari, "PSWG: An automatic stop-word list generator for Persian information retrieval systems based on similarity function & POS information," 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 2015, pp. 111-117.

[12] Masoud Makrehchi and Mohamed S. Kamel, "Automatic extraction of domain-specific stopwords from labeled documents", Proceedings of the IR research, 30th European conference on Advances in information retrieval (ECIR'08), Springer-Verlag, Berlin, Heidelberg, 222-233, 2008

[13] F. Zou et al., "Automatic construction of Chinese stop word list", Proceedings of the 5th WSEAS international conference on Applied computer science, pp. 1009-1014, 2006.

[14] Porter, M.F.: An algorithm for suffix stripping. Readings Inf. Retrieval 4, 313–316 (1980)

[15] Ramanathan, A., Rao, D.: A lightweight stemmer for Hindi. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics(EACL) on Computational linguistics for South Asian Language (2003)

[16] Mishra, U., Chandra, P.: MAULIK: an effective stemmer for Hindi language. Int. J. Comput. Sci. Eng. 4, 711–717 (2012)

[17] Shambhavi, B.R., Kumar, P.R.: Kannada morphological analyzer and generator using trie.Int. J. Comput. Sci. Netw. Secur. 11, 112–116 (2011)

[18] Islam, Z., Uddin, M.N., Khan, M.: A light weight stemmer for bengali and its use in spelling22. Hybrid Approach for Stemming in Punjabi. Int. J. Comput. Sci. Comput. Netw. http://www.ijcscn.com/Documents/Volumes/vol3issue2/ijcscn2013030206.pdf

[19] AbhijitDebbarma, BS Purkayastha, Paritosh Bhattacharya, "Stemmer for Resource Scarce Language using String Similarity Measure", International Conference on Reliability, Optimization and Information Technology (ICROIT), Feb 2014.

[20] Amitava Das, SivajiBandyopadhyay, "Morphological Stemming Cluster Identification for Bangla," Sep 2015.

[21] Brajendra Singh Rajput, Dr.NilayKhare , "A Survey of stemming Algorithms for Information Retrieval," IOSR Journal of Computer Engineering, Vol. 17, Issue 3, PP 76-80, May – Jun. 2015.

[22] Sundar Singh, R.K. Pateriya, "A Survey on various Stemming Algorithms," International journal of computer engineering in research trends, vol. 2, Issue 5, PP 310-315, May 2015.

[23] Sajjad Khan, Waqas Anwar, UsamaBajwa, Xuan Wang, "Template Based Affix Stemmer for a Morphologically Rich Language," The International Arab Journal of Information Technology, Vol. 12, No. 2, Mar 2015.

[24] S.P.Ruba Rani, B.Ramesh, M.Anusha, J.G.R.Sathiaseelan, "Evaluation of Stemming Techniques for Text Classification," International Journal of Computer Science and Mobile Computing, Vol.4 Issue.3, March- 2015.

[25] M.Kasthuri, S.Britto Ramesh Kumar, "A Comprehensive Analyse of Stemming Algorithms for Indian and Non-Indian Languages," International Journal of Computer Engineering and Applications, Vol. VII, Issue III, Sep 2014

[26] Gupta V, "Automatic Stemming of Words for Punjabi Language.", J. (eds) Advances in Signal Processing and Intelligent Recognition Systems. Advances in Intelligent Systems and Computing, vol 264. Springer, Cham, 2014

[27] Ram, V.S., Devi, S.L.: Malayalam stemmer. In: Morphological Analysers and Generators, Mona Parakh, LDC-IL, Mysore, pp. 105–113 (2010).