Text Classification for Marathi Documents using Supervised Learning Methods

Pooja Bolaj
Department of Computer Engineering
PCE, University of Mumbai, India

Sharvari Govilkar
Department of Computer Engineering
PCE, University of Mumbai, India

ABSTRACT
The evolution of Information Technology has led to the collection of large number of text documents. Mostly, researchers worked on English text documents. Today, millions of documents are present in Indian regional languages. So, to classify these documents manually is expensive and time consuming task. Automatic classification can help in better management and retrieval of these documents. From the literature survey, it is found that not much work has been done for classification of Marathi text documents. This paper presents efficient Marathi text classification system using Supervised Learning Methods and Ontology based classification.

Keywords
Text Mining, Support Vector Machine, Naïve Bayes, Modified K Nearest Neighbor, Ontology.

1. INTRODUCTION
The text mining area is gaining more importance because of the availability of large number of electronic documents from different sources; which include unstructured and semi structured information. The main aim of text mining is to enable to users to retrieve information from textual resources and deal with operations like classification (Supervised, Unsupervised and Semi supervised), retrieval and summarization [11]. Data mining, Natural Language Processing (NLP) and Machine Learning Methods work together to automatically classify and discover patterns from the various types of the documents.

Text classification is a process of dividing a given set of documents into one or more predefined classes. This classification of text is done automatically [13]. Usually machine learning techniques are used for automatic text classification. There are mainly two types of techniques namely supervised and unsupervised learning methods. Supervised learning methods assign predefined class label to the testing documents using classification algorithms whereas in unsupervised learning methods grouping of testing documents are done using techniques like clustering.

Marathi is an Indo-Aryan language. It is spoken predominantly by people of Maharashtra. Being morphological rich language; classification of Marathi text documents is difficult task. A Marathi root word consists of many morphological variants with inflections; which makes it difficult to extract feature from Marathi documents.

This paper is organized into 5 sections. Section 1 presents the introduction; section 2 describes Literature Survey of text classification. The proposed system of text classification for Marathi documents is described in section 3. Section 4 presents the methodology and finally the conclusion is included in section 5.

2. LITERATURE SURVEY
Here the relevant literature survey that uses various techniques for text classification is presented. Most of researchers focused on supervised machine learning techniques for classification of Indian regional language documents. These techniques provide better results in the form of accuracy and time efficiency.

Meera Patil, et.al. [1] proposed an efficient system for classifying Marathi text documents Naïve Bayes (NB), Centroid, K-Nearest Neighbor (KNN) and Modified K-Nearest Neighbor (MKNN) classifiers. Then the comparison is done among these four classifiers in terms of accuracy and classification time efficiency. The results show that NB is more efficient for Marathi documents classification in terms of accuracy and classification time; whereas KNN has least accuracy among the four techniques.

Sushma R. Vispute, et al. [2] created an intelligent system for categorizing Marathi documents using LINGO algorithm which is based on Vector Space Model (VSM). It also focuses on providing personalized documents in Marathi language to the end users that are identified from user's browsing history. The comparative analysis shows that VSM performs better than other models such as Boolean model and probabilistic model.

Ashis Kumar Mandal, et. al. [3] explores four promising supervised machine learning methods for categorizing Bangla web documents. The methods include Decision Tree, K Nearest Neighbor (KNN), Naïve Bayes (NB) and Support Vector Machine (SVM). The results show that all the four methods provide satisfactory results with SVM attaining good result in terms of high dimensionality and relatively noisy document feature vector.

K. Rajan, et.al. [4] proposed text classification for one of the Dravidian language i.e. Tamil. This classification is based on Vector Space Model (VSM) and Neural Network Model (NN). The experimental results show that VSM and NN models are effective in classifying and representing Tamil text documents. The accuracy of classifying NN is more compare to VSM.

Abbas Raza Ali, et. al. [5] classified Urdu text documents using statistical techniques such as Naïve Bayes (NB) and Support Vector Machine (SVM). The preprocessing steps include tokenization, normalization, diacritics elimination, stop words elimination and affixes based stemming. The experimental results show that NB classifier is very efficient but has accuracy less than SVM classifier.

Nidhi, et. al. [6] presented for the first time domain based classification of Punjabi text documents using ontology and Hybrid approach (combination of Naïve Bayes and Ontology based classification). They chose Sport domain for creating ontology manually. Their results shows that these approaches

provide better results compared to standard algorithms such as Naïve Bayes classifier (NB) and Centroid classifier.

Kavi Narayana Murthy [7] proposed automatic text classification for Telugu news articles using Naïve Bayes (NB) classifier. The four major categories defined include Politics, Sports, Business and Cinema. The performance of NB is computed in terms of precision, recall and F-measure. The author's technique does not use stop word removal, stemming and morphological analysis.

The review on existing literature reveals that not much work has been carried out for the text classification of Indian regional languages. Some of the supervised learning methods applied include K-Nearest Neighbor (KNN), Modified K-Nearest Neighbor (MKNN), Centroid algorithm, Naïve Bayes (NB), and Support Vector Machine (SVM) on languages like Bangla, Marathi, Tamil, Telugu, Punjabi and Urdu.

Among the classification techniques MKNN, KNN, Naïve Bayes, Centroid and one of the clustering techniques i.e. LINGO algorithm applied on Marathi language. These techniques exclude stop word removal and morphological analysis which would have given better results.

3. PROPOSED SYSTEM

The proposed system is text classification for Marathi documents using supervised learning methods and ontology based classification. The input to the system is Marathi text documents and result i.e. output is classified Marathi documents as per class labels. The classes considered for implementation includes Festival, Sport, Tourism, Literature, Movies etc.

4. METHODOLOGY

The system takes input as set of Marathi language documents. The documents undergo preprocessing steps which include input validation, tokenization, stop word removal, stemming and morphological analysis. Then the features are extracted from preprocessed tokens. Finally supervised machine learning methods and ontology based classification are applied to get output as classified Marathi documents as per class label. The supervised machine learning methods includes Naïve Bayes (NB), Modified K-Nearest Neighbor (MKNN) and Support Vector Machine (SVM).

The proposed approach consists of following phases:

1. Preprocessing
    1) Input Validation
    2) Tokenization
    3) Stop word removal
    4) Stemming
    5) Morphological Analysis
2. Feature Extraction
3. Supervised Learning Methods
4. Output as classified documents

4.1 Preprocessing

4.1.1 Input Validation

First step in preprocessing phase is validating the input documents i.e. Set of Marathi text documents. The input document may contain some words or sentences in other script or language. This step is to analyze whether the input document is valid Devanagari script or not. The words or sentences which are not valid to Devanagari script are simply removed for further processing.

4.1.2 Tokenization

The process of breaking text input into tokens is called tokenization. This tokenization task is possible by searching spaces between the words.

4.1.3 Stop Word Removal

Stop words are the most frequently occurring words which slow down the processing of documents as these words are irrelevant. Hence the removal of stop words enhances the speed of searching by comparing with a corpus of stop words**.**

4.1.4 Stemming

Stemming is important in the system, which uses a suffix list to remove suffixes from words and thus reduces the word to its stem. The result of stemming is stem of word that can be given as input to morphological analyzer for further processing.

4.1.5 Morphological Analysis

The words after stemming are analyzed to check whether they are inflected or not. The aim of morphological analysis is to recognize the inner structure of the word. A morphological analyzer is expected to produce root words for a given input document. The root and stem of word may differ in their forms.

4.2 Feature Extraction

In this step, pre-processor computes feature vector for the input text document using Marathi Dictionary. The feature vector has important features in the text document and their frequencies in the text document. Pre-processor ignores

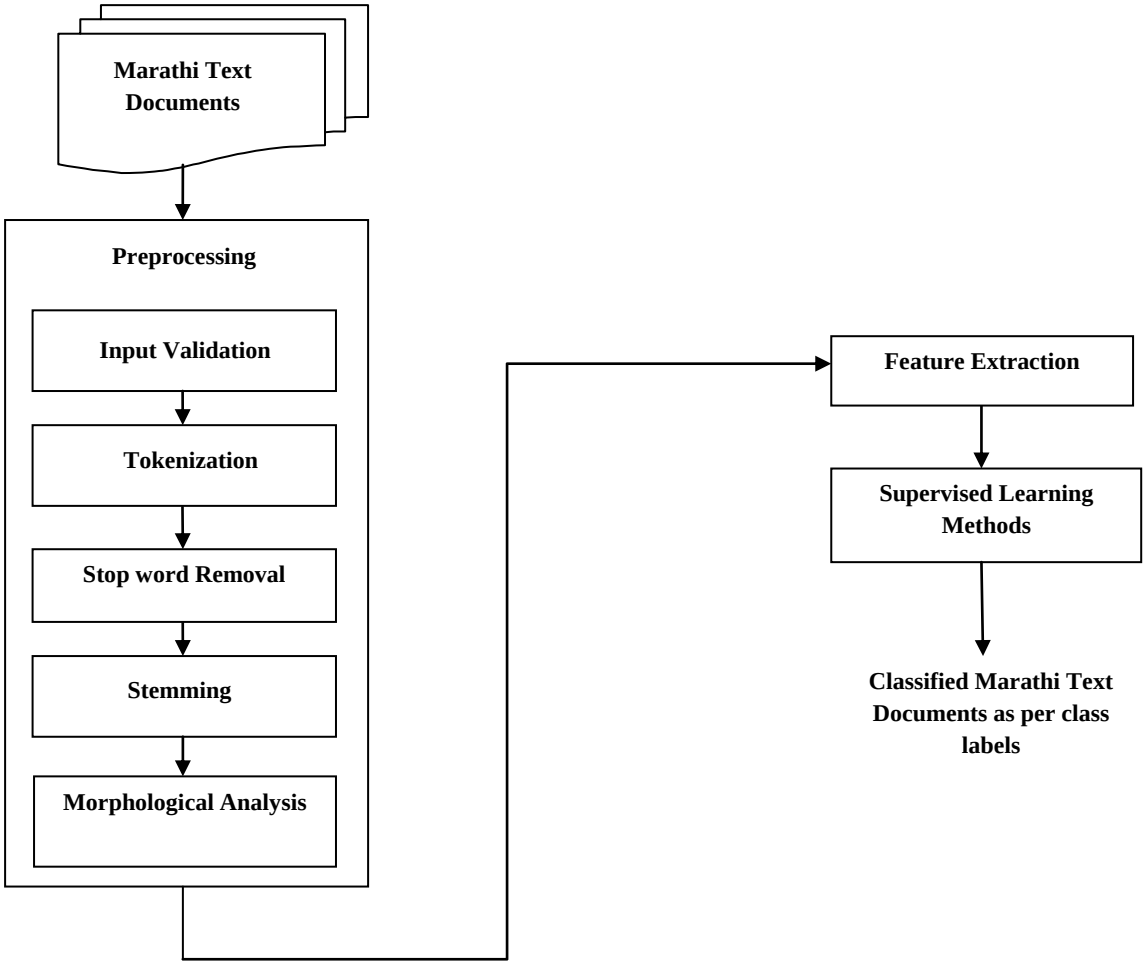

Fig 1: Proposed Architecture

4.3 Supervised Learning Methods

The supervised learning methods and ontology based classification techniques are used for classifying Marathi language documents. The supervised learning methods include Naïve Bayes (NB), Modified K Nearest Neighbor (MKNN) and Support Vector Machine (SVM) of supervised machine learning methods.

4.3.1 Naïve Bayes (NB)

Naïve Bayes is a simple probabilistic classifier based on Bayesian theorem with the assumption of word or feature independence. It means that the word order is irrelevant and consequently that the presence of one word does not affect the presence or absence of another one.

Bayesian Rule is given by

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

For instance, if this applied to spam filter then, let P(C) would be probability of message being spam, P(X|C) is the probability of being given word (input) is spam, Considering the message is spam and P(X) is the probability of a word appearance in a message by using given training data.

Where we could say that:

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidenace}}$$

We will be using *Multinomial Naïve Bayes* which estimates multiple occurrences of term t in a class c. Instead of Boolean Naïve Bayes that estimates occurrences of term t in a class c only once.

The prior probability of class is given as

$$P(C) = \frac{N_C}{N}$$

Where, P(C) is the probability of class and N_c is number of documents of that class over total number of documents indicated by N.

The likelihood of word is given by

$$P(W|C) = \frac{\text{count}(W, C) + 1}{\text{count}(C) + |V|}$$

Where, P(W|C) is likelihood of word of given class, count(W,C) is count of words containing in that class, count(C) is total number of words in class and |V| is vocabulary and 1 is added for smoothing purpose.

Both the training and the testing algorithms are presented below in the form of pseudo code [17]:

TRAINMULTINOMIALNB(C, D)

1. V← EXTRACTVOCABULARY(D)
2. N← COUNTDOCS(D)
3. **for each** c∈C

4. **do** $N_c \leftarrow$COUNTDOCSINCLASS(D,C)
5. prior[c]$\leftarrow N_C/N$
6. $text_c \leftarrow$CONCATENATETEXTOFDOCSINCLASS (D,C)
7. **for each** t€V
8. **do** $T_{ct} \leftarrow$COUNTTOKENSOFTERM($text_c$,t)
9. **for each** t€V
10. **do** condprob[t][c]$\leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11. **return** V, prior, condprob

APPLYMULTINOMIALNB(C, V, prior, condprob)

1. W$\leftarrow$ EXTRACTTOKENSFROMDOC(V,d)
2. **for each** c€C
3. **do** score [c]$\leftarrow$logprior[c]
4. **for each** t€W
5. **do** score [c] +=logcondprob[t][c]
6. **return** argmax $_{c€C}$ score[c]

4.3.2 Modified K Nearest Neighbor (MKNN)

Modified K nearest neighbor classifier is modified version of KNN algorithm. It performs better than KNN algorithm in terms of accuracy. It associates new value, validity with each document in training set. Validity value is directly proportional to stability of the training set document. It computes validity using the topmost nearest neighbors of each training set documents. In this method, H is the numbers of nearest neighbors of each training set document are considered for computing validity of the training set document. Following formula computes validity of each document in training set.

$Validity(x) = 1/H \sum^{H}_{i=1} S(label(x), labelN_i(x))$

S function is as below

$$S(a, b) = 1 \quad a = b$$

$$S(a, b) = 0 \quad a \neq b$$

S function returns value 1 if training document and its i^{th} neighbor in training set both have same label otherwise, the value returned by the function is zero. Validity function is average number of stable neighbors of training set documents. By validity equation, the training set sample that has more stability will have more validity value. Its classification method is similar to KNN. Similarity between testing set document and all training set documents is computed. MKNN is a type of weighted KNN. Following formula computes weighted similarity between testing sample and i^{th} training document,

$W(di, X) = Validity(i) * [Sim(X, D_i) + \alpha]$

α is a parameter. Value considered for it is 0.5. It performs better than KNN since it computes top K neighbors, based on their stability or validity. Following formula computes confidence of X in category c as:-

$$Conf(c, X) = \frac{\Sigma_{(i=1\ldots K|[di.eat]=c)} W(d_i X)}{\Sigma_{i=1\ldots K} W(d_i, x)}$$

Weighted similarities of the documents among the k neighbors which have category c are added and divided by sum of weighted similarities of all k documents to calculate the confidence of X in category c. Algorithm compares confidences in all categories, and assigns the category, for which greatest confidence is found. The notations used by MKNN and pseudo code for MKNN for Marathi text documents are as below. A notable difference between KNN and MKNN is that, KNN does not need any training. MKNN computes validity of each training document in its training phase.

4.3.3 Support Vector Machine (SVM)

The main idea of SVM is to find a hyper-plane that best separates the documents and the margin, distance separating the border of subset and the nearest vector document, is large as possible. The nearest samples of the hyper-plane named support vectors are selected. The calculated hyper-plane permits to separate the space in two areas. To classify the new documents, calculate the area of the space and assign them the corresponding category.

4.4 Ontology Based Classification

Traditional classification methods ignore relationship between words, they consider each term independent of each result. But, in fact, there exist a semantic relation between terms such as synonym, hyponymy etc. Therefore, for better classification results, there is need to understand the context of the text document [6]. The Ontology has different meaning for different users, in this classification task, Ontology stores words that are related to particular domain. Therefore, with the use of domain specific ontology, it becomes easy to classify the documents even if the document does not contain the class name in it. For the proposed system the ontology is built manually.

Steps:

1. Create Domain specific ontology, represented as "bag of words".
2. For each unlabelled document, remove stopwords, punctuations, special symbols and name entities from the document and represent document as "bag of words.
3. To determine in which class unlabelled document belongs, calculate the frequency of document terms matched with class ontology. Assign class to the unlabelled document, if frequency of matching terms with the class ontology is maximum.
4. If no match is found or a document shows same results for two or more classes then that document is not classified into any class, and left for manual classification.

4.5 Output as Classified Marathi Documents

Finally in this phase, the resultant output is set of classified Marathi documents as per the class label. The classes considered are Festival, Sports, History, Literature and Tourism etc. For example, Let's consider two Marathi text documents as input; after preprocessing, applying supervised learning methods and ontology based classification techniques the resultant output is classified Marathi documents belonging to Festival class i.e. Diwali.

5. CONCLUSION

Automatic text classification plays important role in Information Retrieval system. It helps in proper organization and retrieval of data. Not much work has been done on Indian regional language like Marathi. So the proposed system provides text classification of Marathi language documents using supervised learning methods and ontology based classification techniques. In future, the proposed Marathi classification system can be tested with large corpus size and more domains can be added.

6. ACKNOWLEDGMENTS

A Very special thanks to the computer department of Pillai college of Engineering New Panvel for giving us the opportunity to conduct the research. This research paper wouldn't have been possible without the efforts of our principal Dr. RIK Moorthy.

7. REFERENCES

[1] Meera Patil, et. al., "Comparison of Marathi Text Classifiers", ACEEE Int. J. on Information Technology, DOI: 01.IJIT.4.1.4, March 2014.

[2] Sushma R. Vispute, et. al., "Automatic Text Categorization of Marathi Documents Using Clustering Technique", 978-1-4673-2818-0/13, 2013 IEEE.

[3] Ashis Kumar Mandal, et. al., "Supervised Learning Methods for Bangla Web Document Categorization", International Journal of Artificial Intelligence and Application (IJAIA), DOI: 10.5121/ijaia.2014.5508 September 2014.

[4] K. Rajan, et. al., "Automatic classification of Tamil documents using vector space model and artificial neural networks", Expert Systems with Applications 36 (2009) 1091-10918, ELSEVIER, 2009

[5] Abbas Raza Ali, et. al., "Urdu Text Classification", FIT'09, December 16-18, 2009, CIIT, Abbottabad, Pakistan.

[6] Nidhi, et. al., "Domain Based Classification of Punjabi Text Documents using Ontology and Hybrid Based Approach", Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), pages 109-122, COLING 2012, Mumbai, December 2012.

[7] Kavi Narayan Murthy, "Automatic Categorization of Telugu News Articles".

[8] A. Kanaka Durga, et. al., "Ontology Based Text Categorization-Telugu Documents", International Journal of Scientific & Engineering Research Volume 2, Issue 9, September-2011. ISSN 2229-5518.

[9] Nidhi, et. al., "Punjabi Text Classification using Naïve Bayes, Centroid and Hybrid Approach", DOI: 10.5121/csit.2012.2421.

[10] Vishnu Murthy, et. al., "A Comparative Study on Term Weighting Methods For Automated Telugu Text Categorization With Effective Classifiers", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.6, November 2013.

[11] Bijal Dalwadi, et.al., "A Review: Text Categorization for Indian Language", 2349-4476, International Journal of Engineering Technology Management and Applied Sciences, March 2015.

[12] B. Mahalakshmi, et. al., "An Overview of Categorization Techniques", 2249-6645, International Journal of Modern Engineering Research (IJMER). October 2012.

[13] S. Niharika, et. al., "A Survey on Text Categorization", 2231-2803, International Journal of Computer Trends and Technology, 2012.

[14] Monika Dogra, et. al., "A effective stemmer in Devanagari script", Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering -- RTCCE 2013, ISBN: 978-981-07-6184-4 doi:10.3850/ 978-981-07-6184-4_05.

[15] Sharvari S. Govilkar, et. al., "Extraction of Root Words using Morphological Analyzer for Devanagari Script", I.J. *Information Technology and Computer Science*, 2016, 01, 33-39, DOI: 10.5815/ijitcs.2016.01.04.

[16] Dalwadi Bijal, et. al., "Overview of Stemming Algorithms for Indian and Non-Indian Languages", International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 1144-1146.

[17] http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/