ISSN:0975-9646

Automatic Text Categorization of Marathi Language Documents

Aishwarya Sahani, Kaustubh Sarang, Sushmita Umredkar, and Mihir Patil
Department of Computer Science, Pillai College of Engineering, New Panvel, New Mumbai, India.

Abstract— **Information technology generated huge data on the internet. This data is mainly in English language so majority of data mining and natural language processing research work is in English. As the internet usage increased, data in Marathi language also increased. The proposed system presents the document categorization system for Marathi text documents. The system categorizes the Marathi documents and displays it to the end user based on the categories. Similar documents are grouped to form different clusters. The clusters are formed automatically i.e. the system assigns names to the clusters (folders of categorized documents created) based on the content of the documents. Automatic text categorization is useful in better management and retrieval of text documents and also makes document retrieval a simple task. As there has been an increase in digital information available on the internet, there is a growing interest in helping user better find, filter and manage this information.**

Index Terms—**Categorization, Cluster, Feature extraction, Morphological analysis, Marathi documents, News article, Removing inflections using rules, Stemming, Stop words in Devanagari, Suffixes in Devanagari, Tokenization.**

I. Introduction

Text categorization (also known as text clustering) is the task of automatically sorting a set of documents into categories. This task has several applications, including automated indexing of scientific articles according to predefined thesauri of technical terms, filing patents into patent directories, selective dissemination of information to information consumers, automated population of hierarchical catalogues of Web resources, spam filtering, identification of document genre, authorship attribution, survey coding, and even automated essay grading.

Automated text clustering is attractive because it frees organizations from the need of manually organizing document bases, which can be too expensive, or simply not feasible given the time constraints of the application or the number of documents involved.

As the volume of information available on the Internet continues to increase, there is growing interest in helping user better find, filter and manage this information. In past this information is available mainly in English language. But as the Internet usage increased information in languages other than English increased. In Maharashtra (India) state regional language is Marathi. Marathi uses modified version of Devanagari script and is phonetic. This system will help Marathi people for getting the required information in Marathi language. The system will perform the automatic document categorization by retrieving the relevant documents which will reduce user's efforts.

Text categorization is a discipline in NLP and the explosion in the availability of digital information has boosted the importance of such systems, which are nowadays being used in diverse contexts. Although much research isn't done in Marathi language, also, the existing systems aren't as efficient. Thereby, the aim is to build an efficient Marathi text clustering system.

II. Literature Survey Of Similar Work

The result of the 8 regional language papers and 3 English papers about the various algorithms, it has been concluded that LINGO algorithm is the best for Marathi language documents [1]. Also the results can be further improved if morphological analysis is performed on the paper. Naive Bayes which is totally dependent on probability functions is also a good classifier but it needs training to give accurate results. Ontology based algorithms are easy, however the mapping of ontology classes and user classes is an issue as it may lead to ambiguity. Like ontology, K nearest neighbor is also easy to implement and fast but the accuracy of the output is not guaranteed. We also studied VSM where the contents are divided into vectors and the weights of the words are compared with the weight of the keywords. N gram is based on splitting the words into grams and then classifying, although they seem to be language based preprocessing independent but the results may differ if the content is changing.

Thereby, we conclude on the basis of the research that LINGO algorithm proves to be efficient algorithm for Marathi language and thus, LINGO algorithm has been selected.

III. System Architecture

The input to the system would be a set of Marathi documents of different domains. The first unit is Pre-processing unit in which the system will perform Filtration of documents, Script validation, Tokenization, Stop word removal and Stemming and morphological analysis of the tokens. The next unit is Feature extraction followed by LINGO clustering. At the end the output would be sets of clustered documents.

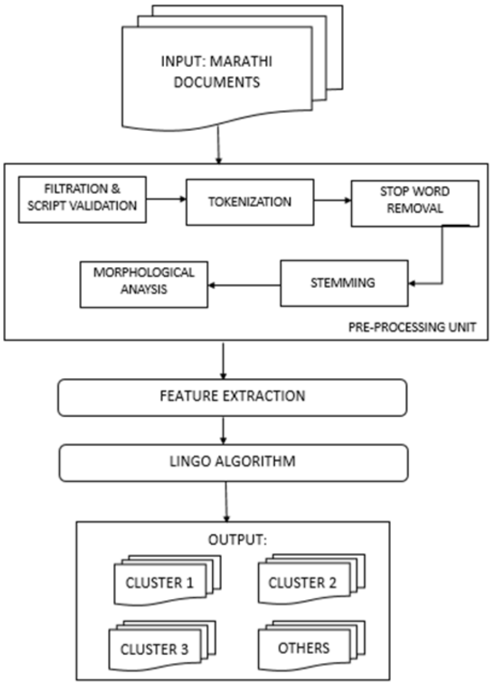

Fig. 1. Architecture of the system

A. Input Documents

The input to the system would be a set of Marathi documents in Devanagari script. The documents can be of any domain like Politics, Entertainment, Literature, Sports, History and Technology.

B. Pre-processing Unit

1. Filtration and Script Validation of Devanagari script

As presence of special characters in Devanagari documents degrades the performance, it needs to be removed. This removal of special characters from Devanagari script is called as filtration of document. Token creation of special characters and its recognition with UTF-8[12] is time consuming which leads to memory wastage. The special characters such as " " ' ' , . / ? [ ] { } : ; \ | ~ ! @ # $ % ^ & * ( ) _ - = + < > are frequently used in many language scripts. Script Validation is concerned with removal of non-Devanagari characters which is done by comparing with the UTF-8 list[12]. These characters will not contribute towards final result.

To perform filtration and script validation operation we have used Unicode values called UTF-8 for Devanagari script document. We compared UTF-8 list with each character of each token, if match found the character is valid and allowed otherwise removed from the document. The aim of this phase is to maintain pure Devanagari script document as input to Morphological Analyzer.

2. Tokenization

Tokenization is a process of converting sentence into a chain of words so that processing word by word can be easily performed. Here we use white space character for tokenization.

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 we remove the stop words to enhance the speed of searching. A corpus of stop words is used to filter out the stop words from the documents.

TABLE I
SAMPLE OF STOP WORDS LIST.

□□□	□□□□	□□□	□□□
□□□□	□□□	□□□□□	□□□□□□□
□□□□□	इतर	□□□	□□□
□□□□	□□□□□□□	□□□□□□□	एवढ
□□□□□	□□□□	□□	□□□□□□□
□□	□□□□	□□□□□□□□	□□□□□□□
□□	□□□□□	□□□□	□□□□□□□□□□
□□□	तर	तरच	□□□□□□□□
□□□□□□□	□□□□□□□	□□□□□	□□□□□
□□□□□□	□□□	□□□□□	□□□□□□
पण	□□□□□	□□□□□	व
□□□	□□□□□□□	□□□□□□	□□□□□□□□□□□

4. Stemming

Suffix stripping is done in this step. The widely used method for this processing is Stemmer which uses a suffix list to remove suffixes from words. The stem is not necessarily the linguistic root of the word. We have designed Corpus of all possible suffixes that occur frequently in the Devanagari script. The corpus is used to remove suffixes from input document.

TABLE 2
SAMPLE OF SUFFIX LIST.

□□□□	□□□□	□□□□	□□□□
□□□□	वर	□□□□	□□□□□
□□□	□□□□	□□□□	□□□□
□□□□	□□□□□□□	□□□□	वर
च	□□	□□□	□□□□□□

5. Morphological Analyzer

We can use following data to get the root for inflected word:

1. List of all the possible suffixes.
2. Rules for inflected words to be replaced by another character.
3. The replacements characters to be made after removal of suffix so that valid root can be formed.

Rule Format:

List of Characters → Replacement Character.

e.g. □□ □□ □□ □□ → म . The meaning of this rule is whenever the word ends with "□□ □□ □□ □□" or has inflection “आ ई उ ए ” are replaced by the character “म ” with inflection “अ ”.

C. Feature Extraction

Intuitively, when writing about something, we usually repeat the subject-related keywords to keep a reader’s attention. Obviously, in a good writing style it is common to use synonymy and pronouns and thus avoid annoying repetition. To be a candidate for a cluster label, a frequent term must:

1. Appear in the input documents at least certain number of times (term frequency threshold).

2. Not begin nor end with a stop word.

A term frequency matrix is the output of this phase where rows are terms and documents are columns and each element indicates the number of times the term has repeated in the document.

D. LINGO Algorithm

LINGO stands for Label Induction grouping. The majority of open text clustering algorithms follows a scheme where cluster content discovery is performed first, and then, based on the content, the labels are determined. But very often intricate measures of similarity among documents do not correspond well with plain human understanding of what a cluster’s “glue” element has been. To avoid such problems Lingo reverses this process—we first attempt to ensure that we can create a human-perceivable cluster label and only then assign documents to it. Specifically, we extract frequent phrases from the input documents, hoping they are the most informative source of human-readable topic descriptions. Next, by performing reduction of the original term-document matrix using SVD, we try to discover any existing latent structure of diverse topics in the search result. Finally, we match group descriptions with the extracted topics and assign relevant documents to them. LINGO algorithm consists of

1. Cluster Label Induction

In this step, the possible number of cluster that can be formed and the candidate label set for the clusters is determined

The phrase with the highest value in the vector is selected as the user graspable concept further by using the cosine ranking the value becomes the score of the candidate of the label of a cluster.

After constructing the term-document matrix where the rows are terms and the columns represent the documents & each item indicates how many times a particular term comes in a particular document. Multiply this idf (Inverse Document Frequency). Denote this matrix as A matrix

Perform SVD of the A matrix, particular vectors of the basis (SVD's U matrix) represent the abstract concepts appearing in the input documents. It should be noted, however, that not all column vectors of the U matrix will be considered in the process of label discovery. Similarly to the SVD-based document retrieval, only the first k basis vectors shall be used and it will be the U_k matrix that will be processed in the further phases.

To calculate the value of k, we will be using the concept of Frobenius matrix norm where the ratio of Frobenius norm of A matrix and the Frobenius norm of the k columns of A matrix exceeds the threshold the value of k is considered as the number of clusters.

Now, we will generate the P (Phrase) Matrix where the rows indicate terms and the columns indicate the phrases and the terms. The matrix indicates how many times the term has appeared in the phrase upon the number of times the term has appeared. The phrase with the highest value in the vector is selected as the user graspable concept further by using the cosine ranking the value becomes the score of the candidate of the label of a cluster.

Algorithm for Cluster Label Induction:

```
a) A term-document matrix of terms not marked as stop-words and with frequency higher than the Term Frequency
b) Compute SVD of A:
c) SVD(A)=∑,U, V;
d) k ← 0; {Start with zero clusters}
e) Repeat
      i. k ← k + 1;
      ii. if(q <$||A_k||_F/||A||_F$) where q is Candidate Label Threshold
      iii. End for false
f) Number of clusters=k
g) P is a phrase matrix which is a matrix of terms and common phrases
h) Select k columns from U matrix, $U_k$
i) do $M=U_k*P$
j) find the largest component mi in each column of M
k) Add the corresponding phrase/term to the Cluster Label Candidates set
l) calculate cosine similarities between all pairs of candidate labels by comparing their respective columns in P
m) identify groups of labels that exceed the Label Similarity Threshold
n) for all groups of similar labels do
o) Select one label with the highest score.
```

p) The phrase matrix multiplied with the Uk matrix. From this matrix, the largest component from every column is selected. The phrase or term that corresponds to the maximum component of the mi vector should be selected as the verbal representation of ith abstract concept. This term can be candidate for the label of a cluster. So, it is added to a set of Candidate Labels.

2. Cluster Content Discovery

The Vector Space Model is used to allocate the given documents to the labels of clusters obtained from the previous phase.

a) Create Q matrix now where each column represents the P matrix columns of the set of Candidate Labels.

b) Calculate $C=Q^TA$, where A is the original term-document matrix.

c) In this way, element cij of the C matrix indicates the strength of membership of the

d) jth document in the ith group.

e) The highest value from every column from the C matrix is selected, the corresponding the row corresponds to the label in which the document is to be added to.

f) Finally, as it is possible that some snippets may match neither of the cluster labels, a special group labeled e.g. "Other topics" can be created in which such snippets should be placed. We can set a threshold for adding to the cluster, if it below that threshold than it is added into "Others" group.

IV. Experimental Results And Analysis

A. Input Dataset:

We have created two sets of input data sets. The two sets are general articles and news articles. Manual data set of 24 documents for General category is created. The documents are in Devanagari script and stored in UTF-8 format. We have considered general categories like Places, Festivals, Entertainment, History, etc.

The second data set consists of 33 documents for News category. We have considered News categories like Sports, Politics, Crime, Economics, Education, Entertainment, Social, etc.

B. Performance Metrics:

The performance metrics is calculated using RAND measure.

The Rand Index is the ratio of pairs of objects correctly clustered out of all possible pairs. This measure estimates the likelihood of an element being correctly classified. This measure is based on the pairwise approach to calculate TP, TN, FP and FN.

Rand Measure = [(TP+TN) / (TP+TN+FP+FN)]*100

C. Experimental Results:

The Rand measure has been calculated for each category i.e. for general and news category for analysis purpose.

1. For General Category Documents:

This is the graph for general category documents. The blue colour shows correctly formed clusters and the red shows incorrectly clustered. The cluster names are automatically assigned by the system based on the documents.

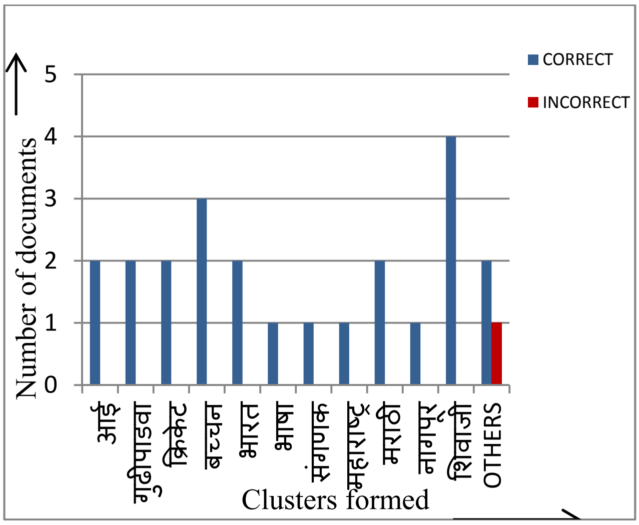

Fig. 2. Cluster of the general documents

The correctly sorted documents for general category:
Rand Measure = [(TP+TN) / (TP+TN+FP+FN)]*100
= (23/24)*100
= 95.83%

2. For News Category Documents:

This is the graph for news category documents.

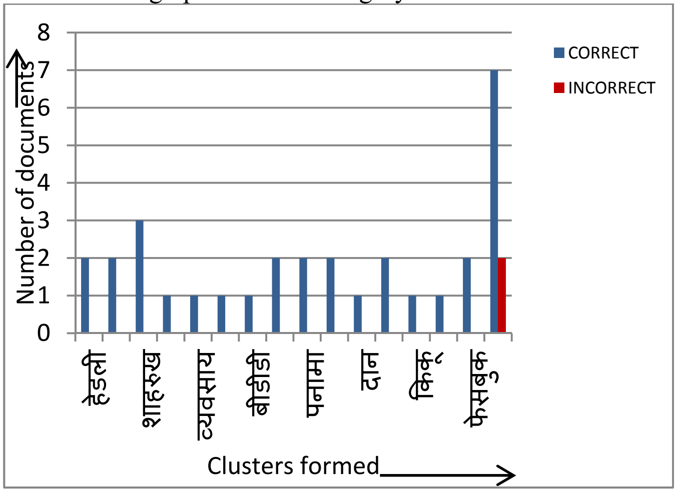

Fig. 3. Cluster of the news documents

The correctly sorted documents for news category:
Rand Measure = [(TP+TN) / (TP+TN+FP+FN)]*100
= (31/33)*100
= 93.93%

Thus, the LINGO Clustering Algorithm is an appropriate algorithm for categorization of Marathi documents as the efficiency attained is much higher than the existing systems. Also, we have modified the algorithm for better results.

V. Applications

A. Hierarchical Categorization of Web Pages

Text categorization has recently aroused a lot of interest also for its possible application to automatically clustering Web pages, or sites, under the hierarchical catalogues hosted by popular Internet portals. When Web documents are catalogued in this way, rather than issuing a query to a general-purpose Web search engine a searcher may find it easier to first navigate in the hierarchy of categories and then restrict a search to a particular category of interest. Clustering Web pages automatically has obvious advantages, since the manual categorization of a large enough subset of the Web is unfeasible.

B. Text filtering

Text filtering is the activity of classifying a stream of incoming documents dispatched in an asynchronous way by an information producer to an information consumer. Typical cases of filtering systems are e-mail filters (in which case the producer is actually a multiplicity of producers), news feed filters, or filters of unsuitable content for Marathi language based Web Content like Marathi Search Engines, Marathi based emails, Marathi Web pages.

C. News Articles Organization

The news dataset can be supplied as input to the system. We can easily get the clusters of similar news. It may be useful to find relevant news regarding a particular incident or to find out what all has been trending in the news. It makes filtering of documents faster and more efficient than manually sorting and finding the relevant articles.

VI. Conclusion

The system yields categorized documents based on the different domains. The categorization is based on the rot words acquired from the input documents. The categorization is done using the LINGO algorithm. This kind of organization may be beneficial in organizations which use Marathi as their primary language and could go a long way in assisting them for document categorization. The segregation is done in matter of seconds rather than the thousands of documents being manually sorted.

The system can be used for Big Data Analytics for clustering web search results for producing for relevant results for the inexperienced as well as the advanced user. The system would cluster all the relevant pages together. Thus, Marathi Search Engines can use the system to cluster the input results and they can be ranked by the label score. The cluster score can be used as a measure to sort clusters in term of importance. The system would be beneficial for newspapers and news channel agencies to sort or organize the huge amount of information that they store. So, headlines or titles in the articles can be given more importance than the other.

References

[1] Jaydeep Jalindar Patil, Nagaraju Bogiri, “Automatic Text Categorization Marathi Documents”, International Journal of Advance Research in Computer Science and Management Studies,Volume 3, Issue 3, March 2015 pg. 280-287, http://ijarcsms.com/docs/paper/volume3/issue3/V3I3-0088.pdf, [Accessed: July19,2015 & time:11:18AM]

[2] Meera Patil, Pravin Game, “Comparison of Marathi Text Classifiers ,ACEEE Int. J. on Information Technology , Vol. 4, No. 1, OMarch 2014, http://searchdl.org/public/journals/2014/IJIT/4/1/4.pdf, [Accessed: July16,2015 & time:11:26AM]

[3] Nidhi, Vishal Gupta, “Domain Based Classification of Punjabi Text Documents using Ontology and Hybrid Based Approach”, Proceedings of COLING 2012: Demonstration Papers, pages 297–304, COLING 2012, Mumbai, December 2012, http://www.aclweb.org/anthology/C12-3037, [Accessed: Aug15,2015 & time:11:09 PM]

[4] Nidhi, Vishal Gupta, “Algorithm for Punjabi Text Classification”, International Journal of Computer Applications (0975 – 8887) Volume 37– No.11, January 2012, http://research.ijcaonline.org/volume37/number11/pxc3876925.pdf , [Accessed: July26,2015 & time:11:26PM].

[5] Ashis Kumar Mandal, Rikta Sen , “Supervised Learning Methods For Bangla Web Document Categorization”, International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 5, No. 5, , September 2014, http://arxiv.org/ftp/arxiv/papers/1410/1410.2045.pdf, [Accessed: Aug17,2015 & time:8:15AM].

[6] Munirul Mansur, Naushad UzZaman and Mumit Khan, “Analysis of N-Gram Based Text Categorization for Bangla in a Newspaper Corpus”, September 2006, http://www.researchgate.net/profile/Mumit_Khan/publication/47506459_Analysis_of_-Gram_Based_Text_Categorization_for_Bangla_in_a_ewspaper_Corpus/links/0c960521f6f72e0b7a000000.pdf?inViewer=true&disableCoverPage=true&origin=publication_detail, [Accessed: Aug17,2015 & time:8:15AM].

[7] Ashraf Odeh, Aymen Abu-Errub, Qusai Shambour, Nidal Turab, “Arabic text categorization algorithm using vector space model”, International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 6, December 2014, http://airccse.org/journal/jcsit/6614ijcsit06.pdf, [Accessed: Aug10,2015 & time:12:52PM].

[8] M Narayana Swamy, M. Hanumanthappa,”Indian Language Text Representation and Categorization Using Supervised Learning Algorithm”, International Journal of Data Mining Techniques and Applications Vol:02, December 2013, Pages: 251-257, http://iirpublications.com/papers/vol2issue2/ijdmta/DEC_13_IJDMTA_004.pdf, [Accessed: Aug10,2015 & time:12:51PM]

[9] Prabin Lama, “Clustering system based on text mining using K-means algorithm”, TURKU UNIVERSITY OF APPLIED SCIENCES THESIS | Prabin Lama,December 2013, http://www.theseus.fi/bitstream/handle/10024/69505/Lama_Prabin.pdf?sequence=1, [Accessed: Aug24,2015 & time:1:14PM].

[10] Maciej Janik, Krys Kochut, “Training-less Ontology-based Text Categorization”, Large Scale Distributed Information Systems Lab (LSDIS) Department of Computer Science, University of Georgia 410 Boyd Graduate Studies Research Center, Athens, GA 30602-7404 ,2011, http://lsdis.cs.uga.edu/~mjanik/JK08-ESAIR08.pdf, [Accessed: Aug24,2015 & time:3:30PM]

[11] Stanislaw Osinski, Jerzy Stefanowski, Dawid Weiss, “Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition”, 2004, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.5370&rep=rep1&type=pdf, [Accessed: Aug24, 2015 & time: 1:25PM].

[12] http://www.unicode.org/charts/PDF/U0900.pdf for UTF-8 Unicode’s used in Devanagari[Accessed: Sept15,2015 & time:12:45PM].

[13]]Elizabeth D. Liddy,”Natural Language Processing”, 2001, http://surface.syr.edu/cgi/viewcontent.cgi?article=1019&context=cnlp , [Accessed: July 16, 2015 & time: 3:30PM]

[14] Fabrizioc Sebastiani, “Text Categorization”, 2005, http://nmis.isti.cnr.it/sebastiani/Publications/TM05.pdf , [Accessed: July16, 2015 & time: 4:15PM].

[15] https://drive.google.com/file/d/0BxnkLjuE3Id_czhmTG1CWFRxWXc/view?usp=sharing for Stop words removal list used in Devanagari [Accessed: Sept17, 2016 & time: 01:20PM].

[16] https://drive.google.com/file/d/0BxnkLjuE3Id_TDlvN0w1Qk8ybG8/view?usp=sharing for Stemming words list used in Devanagari [Accessed: Sept17, 2016 & time: 01:45PM].

[17] Stanislaw Osinski, “An algorithm for clustering of web search results”, 2003, http://project.carrot2.org/publications/osinski-2003-lingo.pdf , [Last Accessed: December23, 2015 & time: 04:34PM].