

## Indian Language Text Representation and Categorization Using Supervised Learning Algorithm

M Narayana Swamy<sup>1</sup>, M. Hanumanthappa<sup>2</sup>

<sup>1</sup>Department of Computer Applications, Presidency College, Bangalore, India

<sup>2</sup>Department of Computer Science & Applications, Bangalore University, Bangalore, India

[narayan1973.mns@gmail.com](mailto:narayan1973.mns@gmail.com), [hanu6572@hotmail.com](mailto:hanu6572@hotmail.com)

### Abstract

*India is the home of different languages, due to its cultural and geographical diversity. The official and regional languages of India play an important role in communication among the people living in the country. In the Constitution of India, a provision is made for each of the Indian states to choose their own official language for communicating at the state level for official purpose. In the eighth schedule as of May 2008, there are 22 official languages in India.*

*The availability of constantly increasing amount of textual data of various Indian regional languages in electronic form has accelerated. So the Classification of text documents based on languages is essential. The objective of the work is the representation and categorization of Indian language text documents using text mining techniques.*

*South Indian language corpus such as Kannada, Tamil and Telugu language corpus, has been created. Several text mining techniques such as naive Bayes classifier, k-Nearest-Neighbor classifier and decision tree for text categorization have been used.*

*There is not much work done in text categorization in Indian languages. Text categorization in Indian languages is challenging as Indian languages are very rich in morphology. In this paper an attempt has been made to categorize Indian language text using text mining algorithms.*

Keywords - Tokens, Lemmatization or Stemming, Stop words, Zipf's law, Vector Space Model, Bayes classifier, k-Neighbor classifier, Decision tree, precision ( $p$ ), recall ( $r$ ), F-measure

### I. INTRODUCTION

Data mining is the main area when dealing with structured data in databases. Text mining refers to the process of analyzing and detecting knowledge in unstructured data in the form of text. The main problem in text mining is that the data in text form is written using grammatical rules to make it readable by humans, So to be able to analyze the text, it first needs to be preprocessed.

There are two fundamental approaches to analyse the text. **First**, Text mining employs Natural Language Processing (NLP) to extract meaning from text using algorithms. This approach can be very successful but it has limitations. **Second**, a different approach using statistical methods is becoming increasingly popular and the techniques are improving steadily [1].

### II. OBJECTIVE

India is the home of different languages. Each state in India has its own official language. The objective of this work is to classify the documents based on language, using supervised learning algorithm. In future, these

categorized documents can be used for summarization.

### III. TEXT REPRESENTATION

The key objective of data preparation is to transform text into a numerical format such as vector space model. To mine text, we first need to process it into a form that data-mining algorithms can use. The Pre-processing steps are shown in figure 1

#### A. Collecting Documents

The work resource for creating this corpus is the World Wide Web itself. The main problem with this approach to document collection is that the data may be of uncertain quality and require extensive cleansing before use.

#### B. Document Standardization

Once the documents are collected, it is common to find them in a variety of different formats, depending on how the documents were generated. The documents should be processed with minor modification, to convert them to a standard format.



Figure 1

### C. Tokenization

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens*, perhaps at the same time throwing away certain characters, such as punctuation. The tokenization process is language-dependent

INPUT	namma dEsha BAрата. nAvu BAratiyaru.		
OUTPUT (Tokens)	namma	dEsha	BAрата
	nAvu	BAratiyaru	

INPUT	ನಮ್ಮ ದೇಶ ಭಾರತ. ನಾವು ಭಾರತಿಯರು.		
OUTPUT (Tokens)	ನಮ್ಮ	ದೇಶ	ಭಾರತ
	ನಾವು	ಭಾರತಿಯರು	

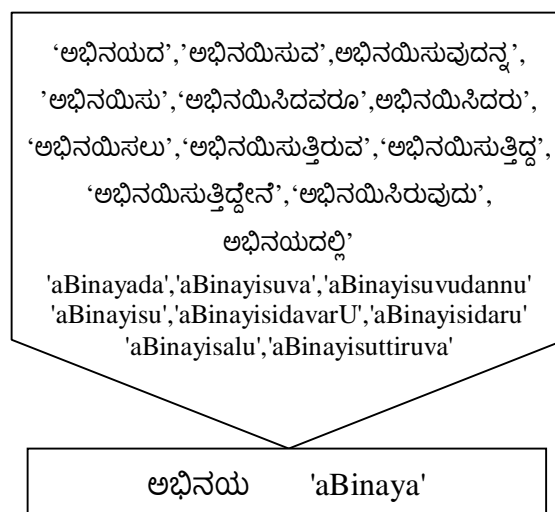
### D. Dropping Common Terms: Stop Words

Some extremely common words are not informative. These words are called *stop words*. The strategy used for determining a

stop list is to sort the terms by *collection frequency* (the total number of times each term appears in the document collection), and then to take the most frequent terms (stop words). These words are discarded during indexing.

### E. Lemmatization

Once tokens are created, the next possible step is to convert each of the tokens to a standard form, a process usually referred to as *stemming* or *lemmatization*. The advantage of stemming is to reduce the number of distinct types in a text corpus and to increase the frequency of occurrence of some individual types.



## IV. PROPERTIED OF CORPUS

The properties of large volume of text are generally referred to as corpus statistics. This data collection comprises 300 documents. The basic statistics of corpus is shown in the table 1.

Language	Kannada	Tamil	Telugu
Documents	100	100	100
Tokens	26315	20360	18427
Vocabulary	20417	15941	14652

The most fundamental property of languages is the one known as Zipf's law. For any language, if we plot the frequency of words versus their rank for a sufficiently large collection of textual data, we will see a clear trend, which resembles a power law distribution. Our experiment on Kannada, Tamil and Telugu corpus statistics is illustrated by Zipf's law.

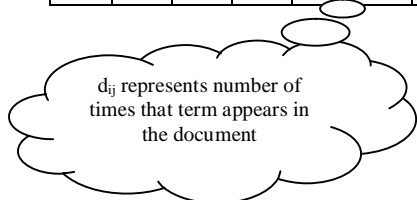
As seen from the table 2, in the low rank extreme of the curve, which are clearly separated from the rest of the words. These are the most frequently used words in our considered data collection.

**V. VECTOR SPACE MODEL**

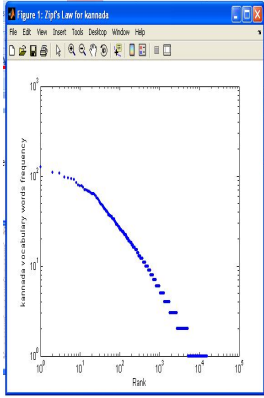
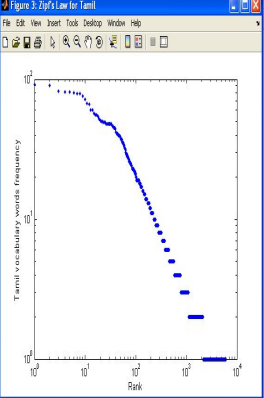
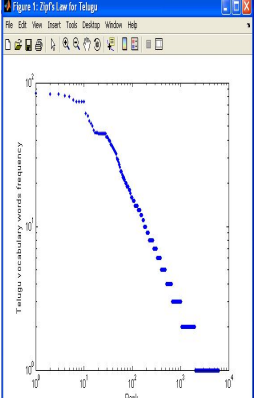
Vector space model or term vector model is an algebraic model for representing text documents as vectors [2]. The principle behind the VSM is that a vector, with elements representing individual terms, may encode a document's meaning according to the relative weights of these term elements. Then one may encode a corpus of documents as a term-by-document matrix  $X$  of column vectors such that the rows represent terms and the columns represent documents. Each element  $x_{ij}$  tabulates the number of times term  $i$  occurs in document  $j$ . This matrix is sparse due to the Zipfian distribution of terms in a language [3]. VSM representation scheme performs well in many text classification tasks [4].

Document-Term Matrix

	T1	T2	T3	T4	T5	T6
D1	4	3	1	3	0	2
D2	0	1	0	2	1	0
D3	2	0	2	0	1	3
D4	0	1	0	0	2	0



Weights are assigned to each term in a document that depends on the number of occurrences of the term in the document. By assigning a weight for each term in a document, a document may be viewed as a vector of weights.

	Zipf's law	Frequent words
Kannada		'sinimA' (ಸಿನಿಮಾ) 'I' (ಈ) 'eMdu'(ಎಂದು) 'citrada'(ಚಿತ್ರದ) 'manaraMjane' (ಮನರಂಜನೆ)
Tamil		'oru' (ಓರು) 'nta' (ಢ್ನತ) 'gkaL' (ಕ್ಕಗ್ಲ) 'paRavaikaLin' (ಪಠಠವಕಗ್ಲಿಢ್ನ) 'kirIccoli' (ಕಿರೀ಑಑ಾಠಿ)
Telugu		'oka' (ಒಕ) 'mariyu' (ಢರಿಯು) 'citraM' (಑ಿತ್ರಂ) 'yokka' (ಯೆಕ್ಕ) 'sinimA'(ಸಿನಿಢಾ)

**A. Term Frequency**

The simplest approach is to assign the weight to be equal to the number of occurrences of term  $t$  in document  $d$ . This weighting scheme is referred to as *term frequency* and is denoted  $tft,d$ , with the subscripts denoting the term and the document in order.

**B. Document Frequency**

Instead, it is more commonplace to use for this purpose the *document frequency*  $dft$ , defined to

be the number of documents in the collection that contain a term  $t$ .

### C. Inverse Document Frequency

we define the *inverse document frequency* (idf) of a term  $t$  as follows:

$$\text{idf}_t = \log \frac{N}{\text{df}_t}$$

Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low.

### D. TF-IDF Weighting

We now combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document. The *tf-idf* weighting scheme assigns to term  $t$  a weight in document  $d$  given by

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

## VI. METHODOLOGY

Document/Text classification plays an important role for several applications especially for organizing, classifying, searching and concisely representing large volumes of information. The *Text Categorization* goal is to label documents according to a predefined set of classes. Classification models describe data relationships and predict values for future observations.

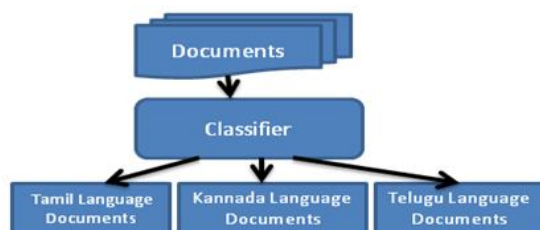


Figure 2

A wide variety of techniques have been designed for text classification, namely decision tree methods, Rule-based classifiers, Bayes classifiers, The nearest neighbor classifier, SVM classifier, regression modeling, neural network classifier and so on. In this paper the decision tree, Naïve Bayes and nearest neighbor Classifier are used for Categorization of the Indian language Documents.

### A. Decision Tree Algorithm(C.4.5)

**C4.5** is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm.

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set  $S = s_1, s_2, \dots$  of already classified samples. Each sample  $s_i$  consists of a  $p$ -dimensional vector  $(x_{1,i}, x_{2,i}, \dots, x_{p,i})$ , where the  $x_j$  represents attributes or features of the sample, as well as the class in which  $s_i$  falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists.

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree telling it to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

### B. Naive Bayes Algorithm

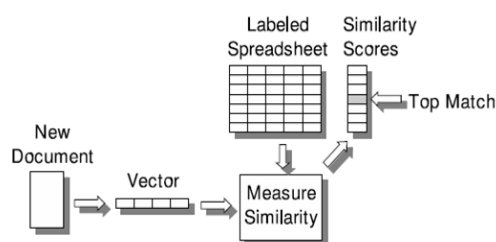
A **naive Bayes classifier** is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. One of the main reasons that NB model works well for text domain because the evidences are "vocabularies" or "words" appearing in texts and the size of the vocabularies is typically in the range of thousands. The large size of evidences (or vocabularies) makes NB model work well for text classification problem [5][6]

NB classifiers can be applied to text categorization in two different ways [7]. One is called *multi-variate Bernoulli model*, and the other is called *multinomial model*. The

difference between these two models stems from the interpretation of the probability  $p(\mathbf{x}|c)$ .

### Nearest Neighbor Algorithm

Finding the nearest neighbors of a document means, to take a new unlabeled document and predict its label. Our documents have been transformed to vector. Each document is now a vector of numbers. Figure 2 is a graphic of the overall process. The new document is embodied in a vector. That vector is compared to all the other vectors, and a score for similarity is computed.



K-Nearest Neighbor is one of the most popular algorithms for text categorization. Many researchers have found that the kNN algorithm achieves very good performance in their experiments on different data sets [8][9][10]

The  $k$ -NN algorithm is a similarity-based learning algorithm that has been shown to be very effective for a variety of problem domains including text categorization [11][12]. Given a test document, the  $k$ -NN algorithm finds the  $k$  nearest neighbors among the training documents, and uses the categories of the  $k$  neighbors to weight the category candidates. The similarity score of each neighbor document to the test document is used as the weight of the categories of the neighbor document. If several of the  $k$  nearest neighbors share a category, then the per-neighbor weights of that category are added together, and the resulting weighted sum is used as the likelihood score of candidate categories. A ranked list is obtained for the test document. By thresholding on these scores, binary category assignments are obtained [13]

### VII. RELATED WORK

Text Categorization is an active and upcoming research area of text mining. Many machine learning algorithms have been applied for many years to text categorization, include decision tree learning and Bayesian learning,

nearest neighbor learning, and artificial neural networks, early such works may be found in[14][15]

The [16] author has reviewed the developments in automatic text categorization over the last decade. Some of the techniques used for automatic categorization have been described. They also mentioned that as far as Indian languages are concerned few result are available. Large scale corpora, good morphological analyzers and stemmers are essential to cope up with the richness of morphology, essential for the Dravidian languages. This paper was the motivation towards work on the South Indian languages.

The author of [17] applied classification algorithm to domain (Sports) Based Ontology for the Classification of Punjabi Text Documents (related to Sports only). Classification Techniques such as kNN technique, Naive Bayes Algorithm, Association Based Classification need Training Set or Labeled Documents to train the classifier to do the classification of the unlabelled documents.

The author of [18] discussed the problem of automatically classifying Arabic text documents. They used the NB algorithm which is based on probabilistic framework to handle our classification problem. Feature selection often increases classification accuracy.

The author [19] used classification algorithm C5.0 to extract the knowledge from Oriya language text documents.

### VIII. DATASET

The corpus is created using three south Indian languages such as Kannada, Tamil and Telugu. We have used 100 documents related to cinema of each language. So, corpus was created using 300 documents. All the documents are cinema related and taken from the WWW. From these documents a corpus is created and its properties are discussed above.

### IX. EXPERIMENTS AND RESULT

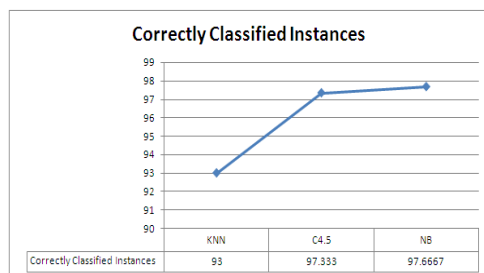
#### A. Algorithm

1. Identify specific language files.
2. Associate a Language label with each of the files.
3. Build a Corpus C
4. Preprocess the Corpus C.
  - a) Apply a Stemming algorithm to reduce all the words to their root form.

5. Generate VSM or a Term Document matrix using Binary Term Occurrence  $D(i, j)$  ( where  $i$  is the document  $i$  and  $j$  is the  $j$ th term of document  $i$ .)

(TF and TF-IDF are not used in the matrix because only the occurrence of the term in the DSL file is relevant for classification; the distinguishing or rarity of the term is irrelevant in this approach)

6. Train the Classifier (kNN,j48 and NB) using  $C$  as training examples.



**B. Evaluation of Text Classifier**

The effectiveness of a text classifier can be

Confusion Matrix								
kNN Classifier			j48 Classifier			NB Classifier		
K	Ta	Tel	K	Ta	Tel	Ka	Ta	Te
an	mi	ug	an	mi	ug	nna	mi	lu
na	l	u	na	l	u	da	l	gu
da			da			da		
87	2	11	99	1	0	10	0	0
2	96	2	1	97	2	2	98	0
4	0	96	4	0	96	5	0	95

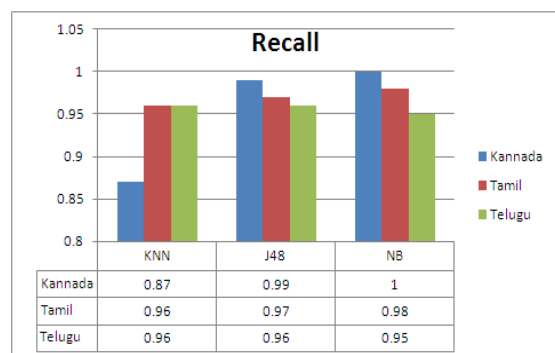
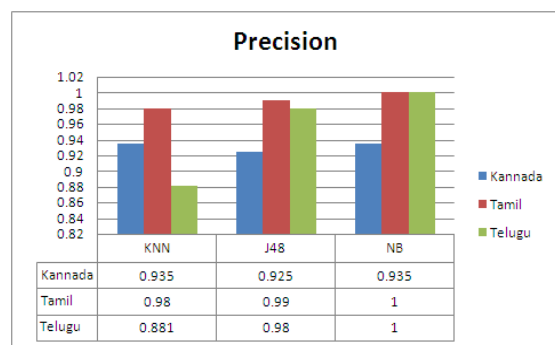
evaluated in terms of its precision ( $p$ ), recall ( $r$ ) and F-measure. . A recall for a category is defined as the percentage of correctly classified documents among all documents belonging to that category i.e. measure of completeness and precision is the percentage of correctly classified documents among all documents that were assigned to the category by the classifier i.e. measure of exactness,. The F- measure combines the two measures in an ad hoc way.

$$\text{precision} = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}},$$

$$\text{recall} = \frac{\text{number of correct positive predictions}}{\text{number of positive class documents}},$$

$$F\text{-measure} = \frac{2}{1/\text{precision} + 1/\text{recall}}.$$

	Precision			Recall			F Measure		
	kNN	J48	NB	kNN	J48	NB	kNN	J48	NB
Kannada	0.94	0.93	0.94	0.87	1	1	0.9	0.97	0.97
Tamil	0.98	0.99	1	0.96	1	0.98	0.97	0.98	0.99
Telugu	0.88	0.98	1	0.96	1	0.95	0.92	0.97	0.97
Average	0.93	0.97	0.98	0.93	1	0.98	0.93	0.97	0.98



**X. CONCLUSION AND FUTURE WORK**

The use of the mining algorithm k Nearest Neighbor (KNN), Naïve bayes and Decision tree C4.5(J48) to south Indian languages such as Kannada, Tamil and Telugu text have been evaluated. We have used a corpus of our own; the corpus consists of 300 documents that belong to 3 categories. All the documents were preprocessed by removing stop words and light stemming all the tokens. The documents were represented using the vector space model.

For measuring the effectiveness of classification algorithm, we used the traditional recall and precision measures The results illustrates that kNN gives 93% accuracy, Decision tree C4.5 gives 97.33% and Naïve Bayes gives 97.66% accuracy. Very satisfactory results have been achieved. It has been proved that text mining algorithm can also applicable for Indian language for

categorization and Naïve Bayes is efficient algorithm for Indian language text categorization.

#### XI. REFERENCES

- [1] Text Mining and Scholarly Publishing, Jonathan Clark, Publishing Research Consortium 2013, PRCTextMiningandScholarlyPublishinFeb2013
- [2] Salton G and McGill M 1983 Introduction to Modern Information Retrieval .McGraw-Hill, New York
- [3] Zipf GK 1935 The Psychobiology of Language. Houghton-Mifflin, Boston, MA.
- [4] Representation and Classification of Text Documents: A Brief Review B S Harish, D S Guru, S Manjunath” IJCA Special Issue on “Recent Trends in Image Processing and Pattern Recognition” RTIPPR, 2010.”
- [5] T. Mitchell. Machine Learning. McCraw Hill, 1996.] [[34] Norbert Fuhr. Probabilistic models in information retrieval. The Computer Journal, 35(3):243–255, 1992.
- [6] Makoto Iwayama and Takenobu Tokunaga. A probabilistic model for text categorization: Based on a single random variable with multiple values. In Proceedings of ANLP-94, 4th Conference on Applied Natural Language Processing, pages 162–167, Stuttgart, DE, 1994. Association for Computational Linguistics, Morristown, US.
- [7] Yang Y. and Liu X., 1999. A Re-examination of Text Categorization Methods [A]. In: Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval [C]. 42-49
- [8] Joachims T., 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features [A]. In: Proceedings of the European Conference on Machine Learning [C].
- [9] Li Baoli, Chen Yuzhong, and Yu Shiwen, 2002. A Comparative Study on Automatic Categorization Methods for Chinese Search Engine In Proceedings of the Eighth Joint International Computer Conference [C]. Hangzhou: Zhejiang University Press, 117-120
- [10] Yang, Y. and Liu, X. (1999). A Re-examination of Text Categorization Methods. In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, pages 42-49.]
- [11] Mitchell, T.M. (1996). Machine Learning. McGraw Hill, New York, NY.].
- [12] Yang, Y. and Liu, X. (1999). A Re-examination of Text Categorization Methods. In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, pages 42-49.
- [13] Lewis and Ringnetto, 1994 D. Lewis, M. Ringnetto, "Comparison of twolearning algorithms for text categorization, “Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), 1994.
- [14] E. Wiener, J. O. Pedersen, and A. S. Zeigend, "A neural network approach to topic spotting," Proceedings of the Fourth Symposium on Document Analysis and Information Retrieval (SDAIR'95), 1995.
- [15] “Advances in Automatic Text Categorization” Kavi Narayana Murthy, Department of Computer and Information Science, University of Hyderabad
- [16] “Punjabi Text Classification using Naïve Bayes, Centroid and Hybrid Approach” Nidhi and Vishal Gupta, Department of Computer Science and Engineering, Panjab University, Chandigarh, India
- [17] “Naïve Bayesian Based on Chi Square to Categorize Arabic Data” Fadi Thabtah, Philadelphia University, Mohammad Ali H. Eljini, AL-Isra Private University, Mannam Zamzeer, University of Jordan, Wa’el Musa Hadi, AL-Isra Private University, Jordan, Communications of the IBIMA, Volume 10, 2009 ISSN: 1943-7765
- [18] “Oriya Language Text Mining Using C5.0 Algorithm” Sohag Sundar Nanda, Soumya Mishra, Sanghamitra Mohanty, P.G. Department of Computer Science and Application, Utkal University, India ISSN:0975-9646
- [19] Andrew McCallum and Kamal Nigam. A comparison of event models for naïve bayes text classification. In Proceedings of AAAI-98 Workshop on Learning for Text Categorization, pages 41–48, 1998.