Automatic Categorization of Telugu News Articles

Kavi Narayana Murthy
Department of Computer and Information Sciences,
University of Hyderabad,
Hyderabad, 500 046,
email: knmcs@uohyd.ernet.in

Abstract

This paper[1] is about automatic text categorization with special emphasis on Telugu. Not much work has been done on Text Categorization in Indian languages so far. In this paper, supervised classification using the Naive Bayes classifier has been applied to Telugu news articles in four major categories totalling to about 800 documents. Category-wise normalized tf-idf are used as feature values. Ten-fold cross-validation has been performed in all cases. Performance obtained is comparable to published results for other languages.

Keywords: Automatic Text Categorization, Telugu Corpus, Naive Bayes Classifier, Machine Learning

1 Introduction

Over the past decade, there has been an explosion in the availability of electronic information. As the availability information increases, the inability of people to assimilate and profitably utilize such large amounts of information becomes more and more evident. The most successful paradigm for organizing this mass of information, making it comprehensible to people, is perhaps by categorizing the different documents according to their subject matter or topic. Automatic text categorization has many applications including indexing for Information Retrieval Systems and Search Engines, Document Organization, Text Filtering (emails, for example), News Aggregation and Organization, and Word Sense Disambiguation.

Not much work has been done on Text Categorization in Indian languages [17, 4, 6, 5]. In this paper we describe the experiments we have conducted on Telugu News Article Corpus developed by us at University of Hyderabad. The corpus has 9870 files totalling to 27.5 Million words (tokens) arising from about 16 Lakh types (distinct word forms). The documents are classified into various categories such as Politics, Sports, Business, Cinema, Health, Editorials, District News, Letters, etc. In this work documents from Politics, Sports, Business and Cinema are selected giving a set of 794 documents. The pur-

[1]The research reported in this paper was supported by the University Grants Commission under the research project entitled "Language Engineering Research" under the UPE scheme

pose is to build a system that can automatically classify a given document into one of these four categories with a high degree of accuracy. We follow a machine learning approach known as the Naive Bayes Classifier. The details of the experiments conducted and results obtained are included.

2 Text Categorization Defined

The aim of Automatic Text Categorization is to classify documents, typically based on the subject matter or topic, without any manual effort. Automated text categorization can be defined as assigning pre-defined category labels to new documents based on the likelihood suggested by a training set of labeled documents. It is the task of assigning a value to each pair $(d_j, c_i) \in DXC$ where D is a domain of documents and C is a set of predefined categories.

There are several variations to this theme:

- Constraints may be imposed on the number of categories that may be assigned to each document - exactly k, at least k, at most k, and so on. In the *single label* case, k = 1 and a single category is to be assigned to each document. If k is more than 1, we have the *multi-label* categorization.

- The text categorization problem can be reduced to a set of binary classification problems one for each category - where each document is categorized into either $c_i or \overline{c_i}$.

- In *Hard categorization*, the classifier is required to firmly assign categories to documents (or the other way around) whereas in *Ranking Categorization*, the system ranks the various possible assignments and the final decision about class assignments is left to the user. This leads us to the possibility of semi-automatic or interactive classifiers where human users take the final decisions to ensure highest levels of accuracy.

- In *Document Pivoted Categorization* a given document is to be assigned category label(s) whereas in a *Category Pivoted Categorization*, all documents that belong to a given category must be identified. This distinction is more pragmatic than conceptual. Thus if all the documents are not available to start with, document pivoted categorization may be more appropriate while category pivoted categorization may be the preferred choice if new categories get added and already classified documents need to be reclassified.

- If only unlabeled training data is available we may have to use unsupervised learning techniques to perform *Text Clustering* instead of classification into known classes.

In this work document pivoted single label hard categorization based on labeled training data has been carried out on Telugu documents.

3 Techniques for Text Categorization

Before the 1990s, the predominant approach to text classification was the knowledge based approach. With the increasing availability of large scale data in electronic form, advances in machine learning and statistical inference, there has been a clear shift over the last decade or so in the approach towards automatic

learning from large scale data. In the Machine Learning approach, a general inductive process (also called the learner) automatically builds a classifier for a category c_i by observing the characteristics of a set of training documents already classified under $c_i or \overline{c_i}$. The inductive process gleans from these labeled training data, the characteristics that a new unseen document should have in order to be classified under c_i. The classification problem is thus an activity of supervised learning.

An increasing number of learning approaches have been applied, including Regression Models [18, 24], Nearest Neighbor Classification [13, 23, 27, 25, 21], Bayesian Probabilistic Approaches [20, 11, 15, 10, 9, 14, 3], Decision Trees [18, 11, 15, 9, 1], Inductive Rule Learning [2, 7, 8, 16], Neural Networks [22, 19], On-line Learning [8, 12], and Support Vector Machines [9]. Yang and Liu [26] have made a systematic comparative study of several of these approaches and concluded that all methods perform comparably when the distribution of documents across categories is more or less uniform.

It has been shown that Naive Bayes Classifier can be used effectively for text categorization in Indian languages [4]. It was observed that better training data where documents are properly classified subject-wise would be highly desirable for further work in this area. In this paper we apply the Naive Bayes classifier to Telugu News Article Corpus developed by us. Performance obtained is comparable to published results for other languages.

3.1 Bayesian Learning Methods

Bayesian Learning is a probabilistic approach to inference based on the assumption that the quantities of interest are governed by probability distributions and the optimal decision can be made by reasoning about these probabilities together with observed data.

In Bayesian Learning methods a maximum *a posteriori* (MAP) probability is computed using the Bayes Theorem. The basic idea is to use the joint probabilities of document terms and categories to estimate the probabilities of categories given a document.

In some cases, the prior probabilities of all the hypotheses are assumed to be uniform and hence bracketed out. This assumption of uniform priors is questionable and has led to criticism of the Bayesian approaches.

Bayesian method requires the estimation of joint probabilities of all the features for each category. In order to simplify this, independence is often assumed. That is, the conditional probability of a feature given a category is assumed to be independent of the conditional probabilities of other features given that category. A Bayesian classifier that makes this independence assumption is termed a Naive Bayes Classifier. The Independence assumption is rarely valid in real world. Yet the method works quite well and is used in practice [11, 15, 10, 3, 14].

To categorize a test document d_j as belonging to a category C_i, the maximum likelihood is estimated over all categories:

$$P(d_j|C_i) = \sum_{w \in d_j} log(P(w|C_i)) \quad (1)$$

The prior probabilities of each category Prior(C_i) are evaluated as the ratios of the number of documents in category C_i to the number of documents in the total collection.

Finally, the posterior probabilities of each category are calculated by adding the log likelihoods to the log priors.

$$P(C_i|d_j) = log(P(d_j|C_i)) + log(Prior(C_i)) \quad (2)$$

A test document is assigned the category with the maximum posterior probability. To minimize misclassification errors due to narrow differences, a threshold value can be used to include a reject option. Performance can then be measured in terms of Precision, and Recall. In order to capture the Precision-Recall trade-off in a single quantity, a combined measure such as the F-measure can be used.

4 Text Representation

Classification systems represent data in terms of a set of features. Feature sets form a compact and effective representation of the whole data. Typically a vector space model is used - each data item can them be visualized as a point in the D-Dimensional feature space where D is the number of features.

In text categorization, each word in a text is a potential feature. In the domain of text categorization, words and word-like features are called *terms*. Documents are treated as bags of terms. Feature dimensions are thus often very large (often running into tens of thousands). *The curse of dimensionality* states that the number of training data samples required grows exponentially with the number of features. Choice of the right subset of potential features is a major concern. A variety of dimensionality reduction techniques are used in pattern recognition but applying them for text categorization requires care. *Stop word removal*, *Morphology or Stemming*, *Identification of Phrases and Collocations* are some of the steps commonly employed in text categorization to obtain more discriminative features and/or to reduce the number of features. It may be noted that these methods are to a large extent language specific.

In this work all the distinct words are taken as features and documents are represented as vectors of these features. No morphology, stemming or stop word removal is employed. This gives us a base line performance. Analysis of the results will help in selection of better features and judicious application of various dimensionality reduction techniques either by statistical methods or through the careful application of linguistic analyses.

4.1 Feature Weighting

Numerical weights need to be computed for the index terms before machine learning techniques can be applied. Here are some of the basic techniques for term weighting:

- *Term Attributes:* Attributes of the terms such as their syntactic categories can be used to weight the terms.
- *Text attributes:* The number of terms in a text, the length of the text etc. can be used.

- *Relation between the term and the text:* Relative frequency of the term in the text, location of the term in the text, relationship with other terms in the text etc.

- *Relation to corpus:* Relation between the term and the document corpus or some other reference corpus can also be used.

- *Expert Knowledge:* Expert knowledge is a potential source but is rarely used.

- *Term Frequency:* Words that occur more frequently in various categories are believed to be more significant in classification into those categories and are thus given higher weightage. Since the occurrence of a rare term in a short text is more significant than its occurrence in a long text, log of the term frequency is used to reduce the importance of raw term frequencies in those collections that have a wide range of text lengths. Anaphoric references and synonyms reduce the true term frequency. In morphologically rich languages, poor morphological analysis or stemming also adds to this effect.

- *Inverse Document Frequency:* Terms that occur in (almost) all documents are useless for classification. The terms that occur in small number of documents are given higher weightage.

- *Inverse Category Frequency:* Inverse Category Frequency could be more appropriate than inverse document frequency since the distribution of documents into categories may be skewed. A log can again be taken to weigh this down so that this weight does not become over-dominating.

- *Product of tf and idf:* Term frequency and Inverse Document Frequency are interrelated. Terms that occur frequently in a particular class but not very frequently in other classes are the most significant. Hence a product of tf and idf is often used.

- *Length Normalization:* Long and verbose texts usually use the same terms repeatedly. As a result, the term frequency factors are large for long texts and small for short ones, obscuring the real term importance. Term frequencies can be normalized for length of texts by dividing them by the total word count in the document, or better still, by the frequency of the most frequently occurring term in the text.

- *Cosine Normalization* Since the directions of the vectors rather than their actual values are considered to be better indicators of the various classes, cosine normalization, where each term weight is divided by a factor representing the Euclidean vector length is often employed.

In this work the normalized tf-idf products are computed category-wise. The tf values are computed based on the frequency of the most frequent term in the document. The feature value for each term w, for category C_i is:

$$P(w|C_i) = \frac{tf_i(w) * log(\frac{N}{n_i(w)})}{\sqrt{\sum_{j=1}^{n}(tf_j(w) * log(\frac{N}{n_j(w)}))^2}} \quad (3)$$

where $tf_i(w)$ is the term frequency for term w in category i, N is the number of documents in the collection, $n_i(w)$ is the number of documents in the category i that include the index term w, and $j = 1..n$ are the categories.

5 Experiments and Results

Not much work has been done so far on Text Categorization in Indian languages [4, 6, 5]. Our experiments with the CIIL/DoE corpus on various Indian languages have not given very good results [4]. In the current work the same technique, namely Naive Bayes classifier has been applied to Telugu News Articles corpora developed by us here. The corpus was developed by downloading the articles from Eenaadu newspaper over 235 days between July 2003 and March 2004 and converting the font-encoded pages into ISCII standard encoding using tools developed by us. The corpus includes 9870 articles totalling to 27.5 Million words. The corpus development and Naive Bayes Classifier software systems have been developed entirely by us here using Perl under Linux.

Of the 9870 documents in the Telugu News Articles corpus, 794 documents in 4 major categories (P-Politics, S-Sports, B-Business, and C-Cinema) have been used in the current set of experiments. The distribution of the documents across these four categories for each language is tabulated below.

Table 1: Distribution of Documents across Categories

Total	Category-wise Breakup			
	Politics	Sports	Business	Cinema
794	307	205	189	93

It may be observed that the distribution of documents in various categories is not uniform - Cinema for example, includes fewer documents.

The performance for the Naive Bayes classifier was evaluated at different threshold values to explore the Precision-Recall trade-off. As the threshold increases, the number of unclassified cases increase. As can be seen from the table below, this results in an increase in precision(P) and a drop in recall(R) up to some point. Beyond that, some documents that were correctly classified may also start getting into the unclassified region resulting in possible a drop in Precision too and in the F-measure. The values given here are averaged over 10 fold cross validation with training and test data selected at random in 80-20 ratio. It may be seen that a threshold of 0.03 gives an increased Precision of about 96 % without significantly affecting the F-Measure.

Table 2: Precision-Recall Trade-off averaged over 10 folds

Threshold	P %	R %	F %
0.00	94.72	94.72	94.72
0.02	95.21	92.58	93.87
0.03	96.03	92.83	94.40
0.04	95.74	91.89	93.76

It can be seen that the performance is as good as those obtained for other languages and much better than our previous results for the CIIL/DoE corpus [4].

Usual clean up techniques such as stop-word removal have not been used. It may also be noted that there is no dictionary and no morphological analysis is done. Inflected and derived forms are treated as separate words and included as features in the system. Performance can be expected to improve further if we employ a morphological analyzer and treat the root

words rather than fully inflected words as features. Telugu is an inflectional and agglutinating language with a number of aspectual auxiliaries, making morphological analysis very complex. As an alternative stemming algorithms can be tried. Even for stemming, vowel harmony and other systematic sandhi changes in the roots/stems need to be considered and simpler, purely statistical methods are unlikely to work well.

6 Conclusions

Not much work has been carried out on text categorization in Indian languages. Here we have described the experiments we have conducted on Telugu documents using Naive Bayes classifier. The results obtained are comparable to published results for other languages. We now have a base system on which a variety of further explorations can be carried out, both from the linguistic point of view and statistical point of view. With the increasing availability of large scale data, affordable memory and computing power, deeper analysis in both linguistic and statistical sense are becoming possible. Morphological analysis and stemming would be high on the agenda. Role of Phrases and Collocations would be worth exploring. Impact of Syntactic Parsing and Word Sense Disambiguation may be explored. Stop word removal and other usual clean up techniques can be incorporated. The continued search for better features and dimensionality reduction techniques would be interesting.

References

[1] C. Apte, F. Damerau, , and S. Weiss. Text mining with decision rules and decision trees. In *Proceedings of Conference on Automated Learning and Discovery*, 1998.

[2] Chidanand Apte, Fred Damerau, and Sholom M. Weiss. Towards language independent automated learning of text categorization models. In *Proceedings of 17th Annual ACM/SIGIR conference*, 1994.

[3] L. Douglas Baker and Andrew K. Mccallum. Distributional clustering of words for text categorization. In *Proceedings of the 21th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 96–103, 1998.

[4] G Siva Charan, Kavi Narayana Murthy, and S Durga Bhavani. Text categorization in indian languages. In R M K Sinha and V N Shukla, editors, *Proceedings of ICSLT-O-COCOSDA - iSTRANS 2004 International Conference - Vol 1*, pages 56–61. Tata McGraw-Hill Publishing Company Ltd, 2004.

[5] Nirmalya Chowdhury and Diganta Saha. An efficient method of feature selection for text document classification. In R M K Sinha and V N Shukla, editors, *Proceedings of ICSLT-O-COCOSDA - iSTRANS 2004 International Conference - Vol 1*, pages 264–267. Tata McGraw-Hill Publishing Company Ltd, 2004.

[6] Nirmalya Chowdhury and Diganta Saha. Unsupervised text document classification using neural networks. In R M K Sinha and V N Shukla, editors, *Proceedings of*

ICSLT-O-COCOSDA - iSTRANS 2004 International Conference - Vol 1, pages 62–68. Tata McGraw-Hill Publishing Company Ltd, 2004.

[7] William W. Cohen. Text categorization and relational learning. In Armand Prieditis and Stuart J. Russell, editors, *Proceedings of ICML-95, 12th International Conference on Machine Learning*, pages 124–132, Lake Tahoe, US, 1995. Morgan Kaufmann Publishers, San Francisco, US.

[8] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–315, 1996.

[9] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[10] D. Koller and M. Sahami. Hierarchically classifying documents with very few words. In *14th International Conference on Machine Learning ICML'97*, pages 170–178, 1997.

[11] David D. Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SIGIR 92*, pages 59–65, 1992.

[12] David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In Hans-Peter Frei, Donna Harman, Peter Schäuble, and Ross Wilkinson, editors, *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, Zürich, CH, 1996. ACM Press, New York, US.

[13] B. Masland, G. Linoff, and D. Waltz. Classifying news stories using memory based reasoning. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

[14] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[15] I. Moulinier. Is learning bias an issue on the text categorization problem? Technical report, 1997.

[16] I. Moulinier, G. Raskinis, , and J.-G. Ganascia. Text categorization: a symbolic approach. In *Proceedings of SDAIR-96, Annual Symposium on Document Analysis and Information Retrieval*, 1996.

[17] Kavi Narayana Murthy. Automatic text categorization. In A R D Prasad, editor, *Proceedings of Semantic Web Workshop - DRTC - ISI - Bangalore*, 2003.

[18] N Fuhr et al. Air/x—a rule-based multistage indexing system for large subject fields. In *Proceedings of RIAO 91*, pages 606–623, 1991.

[19] T Hwee Ng, Wei B. Goh, and Kok L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia, US, 1997. ACM Press, New York, US.

[20] Konstadinos Tzeras and Stephan Hartmann. Automatic indexing based on Bayesian inference networks. In Robert Korfhage, Edie Rasmussen, and Peter Willett, editors, *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 22–34, Pittsburgh, US, 1993. ACM Press, New York, US.

[21] W Lam and C Y Ho. Using a generalized instance set for automatic text categorization. In *Proc. of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 81–89, 1998.

[22] E. Wiener, J. O. Pedersen, and A. S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, 1995.

[23] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of ACM SIGIR*, pages 13–22, 1994.

[24] Y Yang and C G Chute. An example-based mapping method for text categorization and retrieval. In *ACM Transaction on Information Systems (TOIS 94):*, pages 253–277, 1994.

[25] Yiming Yang. An evaluation of statistical approaches to text categorization. pages 69–90, 1999.

[26] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In Marti A. Hearst, Fredric Gey, and Richard Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.

[27] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.