Text Categorization System for English Text Documents using Naïve Bayes Classifier

Kiran Bolaj
Department of CSE, KLESCET, VTU University
Belagavi, India

ABSTRACT
Information technology generated huge data on the internet. Most of this data is mainly in English language. Automatic text categorization is useful in better management and retrieval of these text documents and also makes document retrieval as simple task. Various learning techniques exist for the classification of text documents like Naïve Bayes, Support Vector Machine and Decision Trees, etc. The proposed system uses a Naïve Bayesian method. Bayesian algorithms are often used to classify data in different categories in a way that the systems can be trained and learn from human corrections.

Keywords
Text categorization, Naïve Bayes

1. INTRODUCTION
As internet has huge collection of data in the form of text documents. Categorizing this data is tedious and time-consuming task. Hence, text categorization system is used to classify documents into different categories. Text categorization (also known as text classification or topics spotting) is he task of automatically sorting a set of documents into categories from a predefined set.

Text is cheap, but information, in the form of knowing what classes a text belongs to, is expensive. Categorization of text can provide this information at low cost, but the classifiers themselves must be built with expensive human effort or trained from texts which have themselves been manually classified. Various learning techniques exist for the classification of text documents like Naïve Bayes, Support Vector Machine and Decision Trees etc. The proposed system uses a Naive Bayesian method.

Bayesian algorithms are often used to classify data in different categories in a way that the systems can be trained and learn from human corrections. This system provides a solution based on Bayesian algorithms to classify text in an unlimited number of categories. In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between the features. Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate preprocessing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

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.

2. LITERATURE SURVEY
Several research papers have worked on categorization and automatic content generation of text documents in various languages. El-Kourdi [1] used Naïve Bayes classification method to perform automatic categorization of Arabic web documents with 62% accuracy. Saleh Alsaleem focused on automated text categorization of Arabic web documents using Support Vector Machine (SVM) categorization method with 78% accuracy. Kohilavani [2] and E. Iniya Nehru focused on delivering personalized contents in Tamil Language using Naïve Bayes Classification method with about 89% accuracy. The system used topic analyzer to identify user’s interest and generated personalized content using intelligent evaluator system. Stanislaw Osinski [3] used LINGO Algorithm with the help of Carrot2 framework for categorization of English and Polish documents with 80-95% accuracy.

Meera Patil and Pravin Game [4] have compared the supervised learning methods that include Naïve Bayes, Centroid, K-Nearest Neighbor (KNN) and Modified K-Nearest Neighbor (MKNN) with their algorithms for Marathi text classification. The results show that Naïve Bayes algorithm is most efficient in terms of time and accuracy. The rest of the techniques are time consuming and have less accuracy.

ArunaDevi K., Saveetha R. [5] proposed an efficient method for extracting C-feature for classifying Tamil text documents. Using the C-feature extraction, you can easily classify the documents because C-feature will contain a pair of terms to classify a document to a predefined category.

Nidhi and Vishal Gupta [6] introduced algorithm using ontology-based classification. In first step, remove all special symbols, extra tabs, spaces, and shifts from the text documents then remove stop words. In second step, extract names, places, dates, month names etc. the text document using Gazetter Lists. In third step, calculate Term Frequency (TF) for each remaining word then eliminate term whose term frequency is below the threshold value. In fourth step, calculate Inverse Term Frequency (IDF) of each word and

TFxIDF of each word that are having TF less than threshold value, In the next step, create ontology for each class to its classes. The Gazetter lists are prepared are prepared manual wise classification using ontology based Punjabi text documents. For each remaining eliminate terms whose term frequency is below the inverse document frequency and removing IDF value less than threshold value. In last step, remaining terms from class-wise list, and if maximum terms are matched with one class, assign that class to unlabeled document. They have created sports ontology in Punjabi Language to classify Punjabi sports documents only. The advantage of this ontology is that it doesn't need training data.

3. NAÏVE BAYES CLASSIFIER (NBC)

NBC is a probabilistic classifier of previously unseen data based around the Bayes theorem rule. This rule is one of the most famous theorems in Statistics and is widely used in many fields from engineering and economics to medicine and law.

Naive Bayes Classifier is rather simple algorithm among classification algorithms – other, more complex algorithms giving better accuracy, but if NBC is trained well on a large data set it can give surprisingly good results for much less effort.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes rule is a way of looking at conditional probabilities that allows flipping the condition around in a convenient way. A conditional probability normally written as P(A|B) is a probably that event A will occur, given the evidence B. What makes it "naive" is that you make an assumption that all observed values are independent from each other. For example – in case of natural language, algorithm won't understand the connection between words "delicious" and "cake" which in real world appear close to each other. Nevertheless, as I mentioned above – if system is trained well, results can be much better than you expect.

4. METHODOLOGY

Let's imagine you want to detect language of a document – you will need to calculate the probability that a given document belongs to a given language (class). This probability can be written as follow:

$$p(S|D) = \frac{p(S)}{p(D)} \prod_i p(w_i|S)$$

Where:

$p(S|D)$ is a probability that document D belongs to class S

$p(S)$ is the probability of class S

$p(w_i|S)$ – probability of each token (word) from document appear in class S

$p(D)$ – probability of document D

As you don't need to calculate a precise probability (just only rank classes from highest to lowest probable) and the probability $p(D)$ is always the same, you can drop it and rewrite the rule:

$$p(S|D) = \prod_i \log p(w_i|S)$$

Let's sum up – in order to count score of document D belonging to class S you need to calculate:

$$score_{S|D} = \prod_i \log\left[\frac{count(t_i \ in \ class \ S)}{count(all \ tokens \ in \ class \ S) + count(all \ tokens)}\right]$$

Where t_i is a token (word) from document D. The extra 1 and count (all tokens) is called Laplace **smoothing** and prevent from multiplying by zero. If you didn't have it in any document with an unseen token in it would score zero.

5. ARCHITECTURE

The below fig. shows the architecture of text categorization. The overall architecture is divided into two parts: Front-end and Back-end.

5.1 Front-end

In this, the user is allowed to upload a text file that has to be categorized to a particular category which it belongs from set of categories stored in database.

5.2 Back-end

In this, the user is allowed to upload a text file that has to be categorized to a particular category which it belongs from set of categories stored in database

5.2.1 System to categorize text

This has two phases: Training phase and Classification phase

- **Training phase:** In this phase, for each category file stored by admin, remove stop words, special characters and extract keywords. Calculate the frequency and probability of keywords present in each category file and save it to .srl file which is referred as training dataset.

- **Classification phase:** In this phase, the given input text file is first converted to lowercase. Extract the keywords from the document then calculate the probability of these words from the training dataset. If a probability is higher for particular category, then the given input document is classified into that category.

5.2.2 Database

In text_categorization database there are two tables: categories and admin. Categories table will store the category name, category file path and parent category. Admin table stores username and password to login.

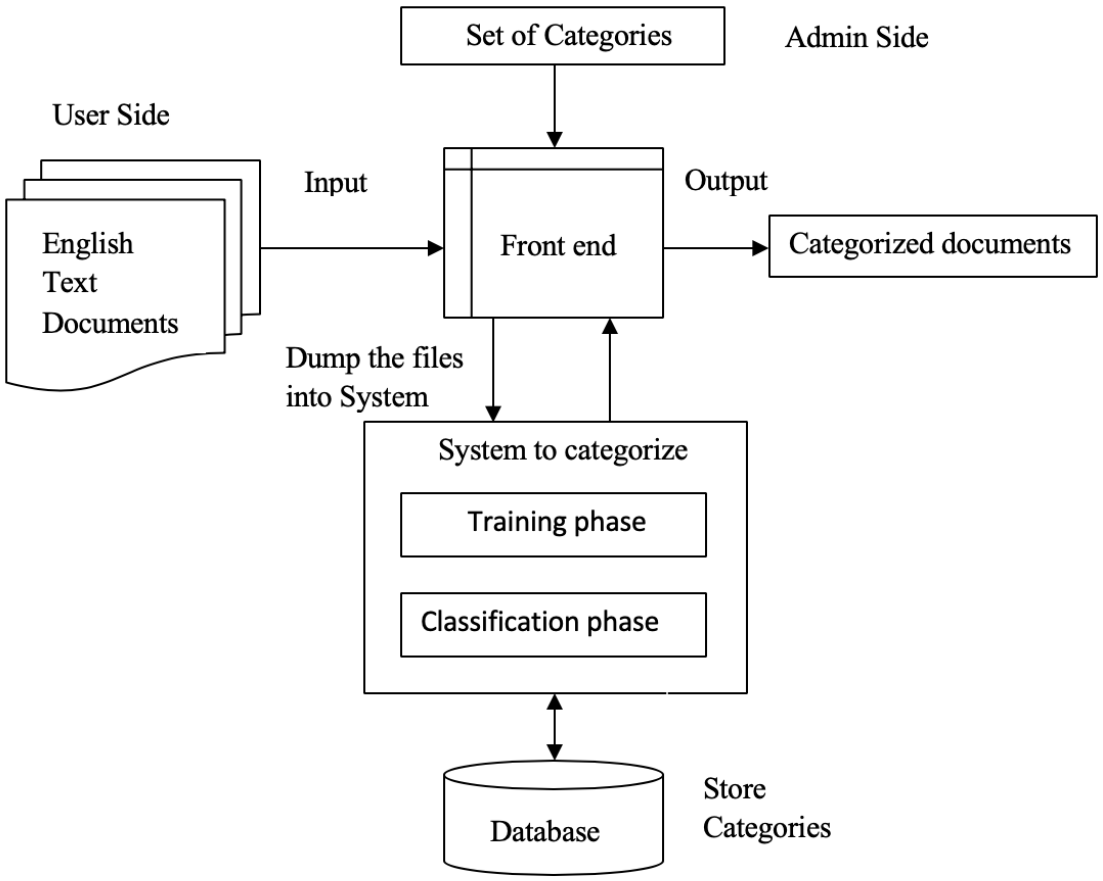


Fig 1: Architecture of Text Categorization System

6. OUTPUT AS CLASSIFIED ENGLISH DOCUMENTS

Finally, in this phase, the resultant output is set of classified English text documents as per the category and sub-category. The categories considered are Technology, Sports, Medical, and Entertainment etc. For example, consider there are two English text documents as input; after applying the Naïve Bayes method, the resultant document will be classified English into Sports and in turn sub-category into Cricket and Entertainment.

7. EXPERIMENT

Consider, you have three different main categories/classes – Sports, Entertainment and Science. And assume you have sub-categories in turn as – Cricket, Bollywood and Medical respectively of each main category.

Below table shows how the Naïve Bayes Classifier works for the given set of test input text document and resultant classified category is shown as output at front end.

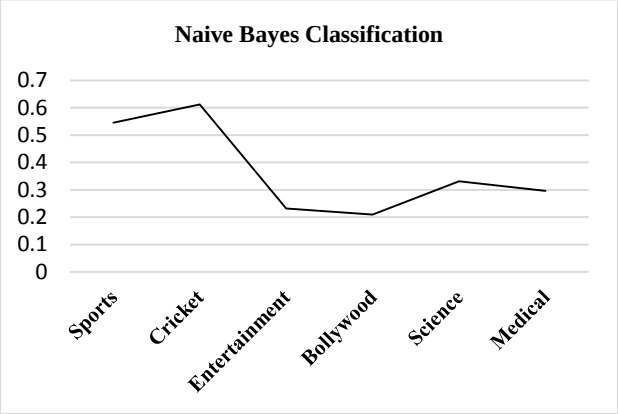
Table1, shows the steps performed to categorize the input document.

Graph1, shows the $score_{s|D}$ for each category.

Table 1. Naïve Bayes Classification

Input document *D* content	Sachin Tendulkar is great Indian Cricket Player.
Training phase	Step1: Calculate count of all the tokens in each class S in the Training set Step2: Calculate count of all the tokens in Training set Step3: Calculate count of tokens t_i from document D Step4: Calculate probability $p(w_i\|S)$ of each token (word) from document appear in class S $p(w_i\|S) = \frac{count(t_i \text{ in class } S)}{count(\text{all tokens in class } S)+count(\text{all tokens})}$
Classification phase	Step4: For each class S, calculate the probability $p(S\|D)$ of a D being in class S, $p(S\|D) = \prod_i \log p(w_i\|S)$ Step5: Calculate the maximum of all the probabilities $p(S\|D)$ of a D being in class S.
Output	**Category:** Cricket

Graph2. $Class\ S$ versus $score_{s|D}$



8. FURURE SCOPE AND APPLICATIONS

8.1 Future scope

The scope of future work can also deal with Incremental learning, which stores the existing model and processes the new incoming data more efficiently. In the field of analyzing the subjectivity of the sentiments/opinions, advanced preprocessing techniques involving the removal of superfluous and malapropos data in the corpus will enhance the information retrieval.

The image classification model can be enhanced in future, by including more low-level features such as shape and spatial location features apart from optimizing the weights and learning rate of the neural network. Self-Organizing Feature map that considers training and mapping which automatically classifies the new input vector could be designed for better classification with neural networks.

8.2 Applications

1. Document Organization
2. Text Filtering
3. Word Sense Disambiguation
4. Hierarchical Classification of Web Pages
5. Spam Filtering
6. Automatic Survey Coding
7. Sentimental Analysis
8. Ontology based classification
9. Opinion Analysis model

9. CONCLUSION

The rapid development of the Information technology has led to the collection of documents in English languages. To classify millions of documents manually is an expensive and time-consuming task. Therefore, text categorization systems are constructed which sort a given set of documents into different classes and whose accuracy and time efficiency is much better than manual text classification**.** In this project, as demonstrated the requirement of improving the efficiency of classification techniques based on Naïve Bayesian, which is a good machine learning algorithm. The results show that the approach to classify English text document is a reasonable and effective one. However, there are lots of enhancements to be done in future. This Naïve Bayes classifier can also be used to classify the text documents of other Indian regional languages like Telugu, Kannada, Marathi, Tamil, Punjabi, Bangla, etc.

10. REFERENCES

[1] El-Kourdi M., Bensaid A. and Rachidi T., "Automatic Arabic Document Categorization Based on the Naive Bayes Algorithm", Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages, pp. 51-58, August 2004.

[2] Kohilavani, S., Mala, T., Geetha, "Automatic Tamil Content Generation IAMA2009, IEEE International Conference, Sep 2009.

[3] Stanislaw Osinski, "An algorithm for clustering of web search results ", Master's thesis, Poznan University of Technology, Poland, 2003.

[4] Meera Patil, Pravin Game, "Comparison of Marathi Text Classifiers", *ACEEE Int. J. on Information Technology*, DOI: 01. IJIT.4.1.4, March 2014.

[5] ArunaDevi, K., Saveetha, R., "A Novel Approach on Tamil Text Classification Using C-Feature", 2321-0613, 2014. *IJSRD International Journal of Scientific Research & Development,* 2014.

[6] Nidhi, Vishal Gupta, "Punjabi Text Classification using Naïve Bayes, Centroid and Hybrid Approach", DOI: 10.5121/csit.2012.2421.

[7] Savita P. T., Santoshkumar B., "Effective Email Classification for Spam and Non-Spam", *International Journal of Advanced Research in Computer Science and Software Engineering*, June 201.