Improving Accuracy of Text Classification for SMS Data

Hiral Padhiyar[1] Prof. Purvi Rekh[2]
[1,2]Department of Computer Engineering
[1, 2] Uka Tarsadia University, India

Abstract— Text classification has become one of the major techniques for organizing and managing online information; similarly SMS classification is also an important task now a day. In this paper, we have focused on the issue of short words used in SMS (hpy for happy, bday for birthday) which reduces classification accuracy, so after removing such words with original words, we got better accuracy. We used Decision tree Algorithm for classification of SMS data as it is giving better accuracy then other classifiers. But still replacing all possible short words for the given word dynamically by the original word is an issue.

Key words: Decision Tree, Text Classification

I. INTRODUCTION

Data mining refers to extracting or "mining" knowledge from large amounts of data. There are different types of techniques for mining the data. One of its techniques is classification. Classification can be described as a supervised learning algorithm in the machine learning process [1]. Data classification and association rule mining are two well-known tasks of data mining. Data classification task is simply defined as the organization of data in predefined classes. This technique is mainly based upon the set of input and output, which is a supervised technique. In order to classify the document the set of input and output examples are used to train the model, which is being used [2]. Text classification is the process of classifying documents into predefined categories based on their content. Several methods have been used for text classification such as: Naïve Bayes, Support Vector Machines, K-Nearest Neighbor, Artificial Neural Networks, Decision Trees, and Statistical Classifiers. It assigns class labels to data objects based on prior knowledge of class which the data records belong.

In classification a given set of data records is divided into training and test data sets. The training data set is used in building the classification model, while the test data record is used in validating the model. The model is then used to classify and predict new set of data records that is different from both the training and test data sets. Supervised learning algorithm (like classification) is preferred to unsupervised learning algorithm (like clustering) because its prior knowledge of the class labels of data records makes feature/attribute selection easy and this leads to good prediction/classification accuracy.

Because of the daily explosive increase in the volume of available electronically information on the Internet, companies, intranets, as well as other media, there is an urgent need to effectively manage such information in order to help users to retrieve them efficiently [4]. Managing and filtering such amount of data, especially textual data, is a complex task. Data mining provides several techniques to solve the problems related to textual data like classification, association mining, characterization, and clustering.

Text Categorization (TC) has become one of the major techniques for organizing and managing online information. Data classification task is simply defined as the organization of data in predefined classes. As defined by several researchers, Text Categorization (or Text Classification) is the process of automatically predicting the valid categories of text documents. Formally, text classification is the problem of best assigning pre-defined class labels to incoming unclassified text documents where class labels are defined based on a sample of pre-classified text documents.

Generally, automatic text categorization has been used in several applications and different domains such as digital library systems, document management systems, search engines, electronic e-mail filtering systems, newsgroups classification systems, information filtering and routing, and survey data grouping [4].

The classification method is usually divided into two major steps; classification model construction and model validation or testing. Each step includes several additional sub steps.

The overall methodology consists of two major phases: Text pre-processing, and Text classification. The first phase deal with model construction, while the second represents the classification algorithm used for the prediction.

II. PROCESS OF TEXT CLASSIFICATION

To classify the text documents first pre-processing is applied as shown in below figure. This process aims to clean the text and reduce the document space or Data pre-processing reduces the size of the input text documents significantly [3]. It includes four steps:

1) Tokenization
2) Stop word removal
3) Stemming
4) Terms selection

All text documents go through a pre-processing phase. The training data set goes through all steps of the pre-processing phase, while the documents to be classified (testing data) go through the first three steps only.

A. Tokenization

To perform the classification process correctly, the tokenization process is necessary. The text tokenization process is performed on all input documents. Each text document is segmented into words (terms) which are known as token that form the document [4].

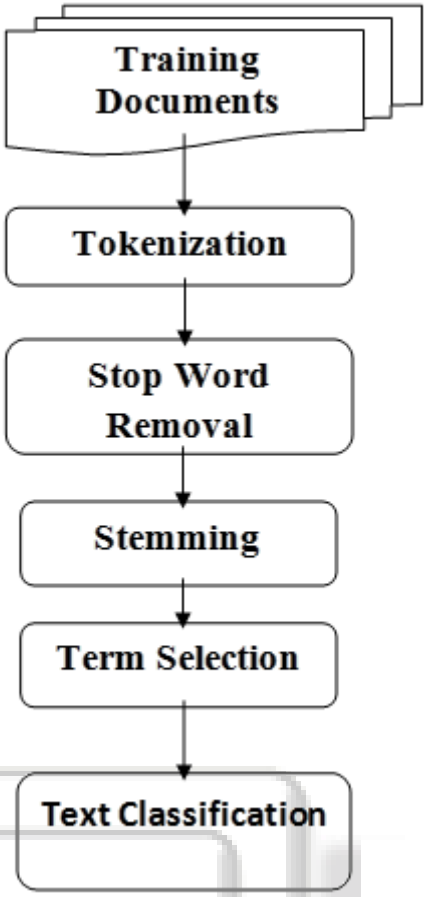

Fig. 1: process of text classification

B. Stop word removal

Stop-words are functional words which occur frequently in the language of the text (for example, 'a', 'the', 'an', 'o' etc. in English language), so that they are not useful for classification. So, the less important or meaningless words are eliminated. It is known that each document, irrespective of its type, includes certain words that are sometimes frequent and do not provide precise discrimination about a document content [4]. These types of words cannot be used to describe the classes that a document belongs to.

C. Stemming

Stemming is the process of segmenting and separating affixes from a word to produce a word stem. This process is used to reduce the term set size that represents each document in the collection. In this stemming algorithm each letter is assigned a weight value that ranges from 0 to 5 [4].

D. Terms selection

After the stop words removal and stemming steps are performed, the documents are represented using the vector space model. Vector space models are based on the basic assumption that a document can be represented as a vector regardless of the order of words [4]. This representation is able to retain enough useful information about the document. In this model each term in a given document is assigned a weight value based on the used weighting scheme, and then the documents are represented as a vector of weighted terms.

As shown in [5], the contribution of each word are inconsistent in the text, important words can represent the core content of the text. The contribution of words is reflected in its weight, this article defines the weight of words from the following three aspects.

1) As for the frequency of words in the text, certainly the more frequently words appear, the more contributions words have for the text.
2) The frequency of the word in the text is high and appears not only in the same type texts but all types of texts.
3) These words appearing in the different position have different contributions for the text; of course, the words appearing in the title bar make more contributions than in content.

When all terms in a given document are assigned weight values, such values are then used to select the important terms. Terms that have a weight value that is higher than or equal to a given threshold value defined by the user are retained to represent this document.

Several techniques can be used to determine the weight of terms in a document, however; most used methods are based on the importance and the appearance of terms in the document and in the collection of documents called the term frequency (TF) and inverse document frequency (TF-IDF). TF-IDF is considered one of the most popular methods that are used to compute the term weight.

Term frequency (TF) represents the number of times that a term *i* appeared in a document *j*, whereas the inverse document frequency is a discriminating measures for a term *i* in the document collection. The inverse document frequency (IDF) of a term *i* in a document *j*, is computed as in Equation.

$$IDF_{i,j} = \log (N/n_i)$$

Where:

N: is the total number of documents in the collection.

n_i: is number of documents in the collection that contain term *i*.

After the term frequency and inverse documents frequency are computed for each term, the term weight is obtained using Equation.

$$W_{i,j} = TF_{i,j} \times IDF_{i,j}$$

Where:

$TF_{i,j}$: Term Frequency of term *i* in document *j*.

$IDF_{i,j}$: Inverse Documents Frequency of term *i* in document *j*.

III. TEXT CLASSIFICATION

To validate the generated associative classifier, another data set; called the testing data that differs from the training data set, is used in this process. If the classification accuracy is acceptable, this model can be used for future predictions. Several methods can be used for classification using a set of classification rules. Such classification algorithms are Naive Bayes, Decision Tree and Support Vector Machine, etc.

Performance comparison of algorithms is given in below table 3.1.

Algorithm	Accuracy

Naïve Bayes	65.00%
Decision Tree	83.33%
SVM	63.33%

Table. 1: Performance Comparison of Algorithms

Decision tree algorithm is a data mining induction techniques that recursively partitions a data set of records using depth-first greedy approach or breadth-first approach until all the data items belong to a particular class. Decision tree algorithm is a data mining induction techniques that recursively partitions a data set of records using depth-first greedy approach or breadth-first approach until all the data items belong to a particular class. The decision tree consists of nodes that form a *rooted tree*, meaning it is a *directed tree* with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an *internal* or test node. All other nodes are called leaves (also known as terminal or decision nodes). The tree structure is used in classifying unknown data records. At each internal node of the tree, a decision of best split is made using impurity measures [1].

IV. RESULT ANALYSIS

We have applied both algorithms on some text documents. Result of this algorithm was in form of accuracy of text. From these algorithms decision tree algorithm gave best accuracy.

Accuracy is defined as [1]:

$$\text{Accuracy}= \frac{2(recall \times precision)}{recall + precision}$$

Where,

Recall = this is the percentage of documents that are relevant to the query and were retrieved.

It is formally defined as

$$\text{Recall}= \frac{|\{relevant\} \cap \{retrieved\}|}{|\{relevant\}|}$$

Precision = this is the percentage of retrieved documents that are in fact relevant to the query.

It is formally defined as

$$\text{Precision} = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{\{retrieved\}\}|}$$

Here We applied both algorithm on same text documents two times but there is some difference in text and that difference is that in original text some short-form of words are used. First, we find accuracy on this original text document and then we replace those short-forms into its original words (i.e. short-form of birthday is bday).

The test data set used for classification:

Class Label	Text document
BDAY	B1,...,B30
NEWYEAR	N1,..,N30

Table. 2: Dataset

Comparison between algorithms by increasing sms text files:

No. of text files	Naive Bayes	Decision Tree	SVM
5	30.00%	80.00%	20.00%
10	75.00%	90.00%	70.00%
15	66.67%	73.33%	63.33%
30	65.00%	83.33%	70.00%

Table. 3: Comparison between Algorithms

After this analysis we found that Decision tree gave better accuracy even for small amount of Training Data.

Comparison of results after replacement and before replacement of short words is shown in below table.

No of text files	Accuracy before replacement	Accuracy after replacement
5	80.00%	80.00%
10	90.00%	90.00%
15	73.33%	76.67%
20	82.50%	90.00%
30	83.33%	95.00%

Table. 4: Comparison of Results

V. CONCLUSION AND FUTURE WORK

In this paper, we focused on the issue of short words used in SMS which affects classification accuracy, so we replaced such short words with original words and found increment in classification accuracy. For SMS classification, we compared classification algorithms like SVM, Naive Bayes and Decision tree, out of which Decision tree algorithm gave better accuracy than the other algorithms. So we used Decision tree algorithm for classification of SMS data before and after replacement of short words and found increment in accuracy. Still replacing all possible short words (like bdy, bday etc for birthday, hpy, hapy etc for happy) dynamically is an issue.

REFERENCES

[1] Kamber, J. H. Data Mining: Concept and Techniques.
[2] M.Sukanya, S. (2012). Techniques on Text Mining. ICACCCT.
[3] Mita K. Dalal, M. A. (2011). Automatic Text Classification: A Technical Review. International Journal of Computer Applications .
[4] QASEM A. AL-RADAIDEH, E. M.-S. (2011). An Approach for Arabic Text Categorization Using Association Rule Mining. International Journal of Computer Processing Of Languages .
[5] Yun Yang, Y. W. (2010). The Improved Features Selection for Text Classification. 2nd International Conference on Computer Engineering and Technology. IEEE.

