Preprocessing Techniques in Text Categorization

Pritam C. Gaigole (*), L. H. Patil (**), P.M Chaudhari(**)
(*) Department of CSE,Priyadarshini Institute of Engineering& Technology, Nagpur
(**)Department of CSE, Priyadarshini Institute of Engineering & Technology, Nagpur
(**)Department of IT, Priyadarshini Institute of Engineering & Technology, Nagpur

ABSTRACT

Bulk data is generated in the era ofInformation Technology. If it is not stored in aproperly systematic manner then the generated datacannot be reused. This is because navigation becomes
if not impossible, certainly very difficult. The data generated is to analyze so as to maximizethe benefits, for intelligent decision making. Textcategorization is an important and extensively studiedproblem in machine learning. The basic phases in textcategorization include preprocessing features, extractingrelevant features against the features in a database, andfinally categorizing a set of documents into predefinedcategories. Most of the researches in text categorization arefocusing more on the development of algorithms andcomputer techniques.

Keywords

Preprocessing, Text categorization

I. INTRODUCTION

Text categorization is the problem of automatically assigning predefined categories to free text documents, while more and more textual information is available online, effective retrieval is difficult without good indexing and summarization is one solution to this problem. A growing number of statistical classification methods and machine learning techniques have been applied to text categorization in recent years, including multivariate regression models[13] nearest neighbor classification[12], Bayes probabilistic approaches[15], decision trees[15], Neural networks[2], Symbolic rule learning[10] and Inductive learning algorithms[11].

A major characteristic or difficulty of text categorization problems is the high dimensionality of the feature space. The native feature space consists of the unique terms(words or phrases) that occur in documents, which can be tens or hundreds of thousands of terms for even a moderate sized text collection. This is prohibitively high for many learning algorithms, few neural network, for example, can handle such a large number of input nodes. Bayes belief models as another example, will be computationally intractable unless an independence assumption(often not true) among features is imposed. It is highly desirable to reduce the native space without sacrificing categorization accuracy. It is also desirable to achieve such a goal automatically, i.e. no manual definition or construction of features is required.

Automatic feature selection methods include the removal of non-information terms according to corpusstatistics, and the construction of new features which combine lower level features (i.e. terms) into higher level orthogonal dimensions.

While many feature selection techniques have been tried, through evaluations are rarely carried out for large text categorization problems. This is due in part to the fact that many learning algorithms do not scale to high dimensional feature space. That is if a classifier can only be tested on a small subset of the native space, one cannot use it to evaluate the full range of potential of feature selection methods. A recent theoretical comparison. For example, was based on performance of decision tree algorithm in solving problems with 6 to 180 features in the native space [14]. An analysis on this scale is distant from the realities of text categorization.
Amazing development of Internet and digital library hastriggered a lot of research areas. Text categorization is oneof them. Text categorization is a process that group textdocuments into one or more predefined categories basedon their contents [1]. It has wide applications, such as emailfiltering, category classification for search engines anddigital libraries. Associative text classification, a task thatcombines the capabilities of association rule mining andclassification, is performed in a series of sequentialsubtasks. They are the preprocessing, the association rulegeneration, the pruning and the actual classification. Out ofthese, the first step, that is, 'Preprocessing', is the mostimportant subtask of text categorization.The importance of preprocessing is emphasized by the factthat the quantity of training data grows exponentially withthe dimension of the input space. It has already beenproven that the time spent on preprocessing can take from50% up to 80% of the entire classification process [2],which clearly proves the importance of preprocessing intext categorization process.This paper discusses the various preprocessing techniquesused in the present research work and analyzes the effect ofpreprocessing on text categorization using machinelearning algorithms. Section 2 gives an overview of thework in text preprocessing. Section 3 explains thepreprocessing steps used.

II. TEXT PREPROCEESING

The preprocessing phase of the study converts the originaltextual data in a data-mining-ready structure, where themost significant text-features that serve to differentiatebetween text-categories are identified. It is the process ofincorporating a new document into an informationretrieval system. An effective preprocessor represents thedocument efficiently in terms of both space (for storing thedocument) and time (for processing retrieval requests)requirements and maintain good retrieval performance(precision and recall). This phase is the most critical andcomplex process that leads to the representation of eachdocument by a select set of index terms. The mainobjective of preprocessing is to obtain the key features orkey terms from online news text documents and to enhancethe relevancy between word and document and therelevancy between word and category.

III.PREPROCESSING STEPS

The goal behind preprocessing is to represent eachdocument as a feature vector, that is, to separate the textinto individual words. In the proposed classifiers, the textdocuments are modeled as transactions. Choosing thekeyword that is the feature selection process, is the mainpreprocessing step necessary for the indexing ofdocuments. This step is crucial in determining the quality of the nextstage, that is, the classification stage. It is important toselect the significant keywords that carry the meaning, anddiscard the words that do not contribute to distinguishingbetween the documents.

3.1 Stop Word Removal

Many of the most frequently used words in English areuseless in Information Retrieval (IR) and text mining.These words are called 'Stop words. Stop-words, whichare language-specific functional words,are frequentwordsthat carry no information (i.e., pronouns, prepositions,conjunctions). In English language, there are about 400-500 Stop words. Examples of such words include 'the', 'of','and', 'to'. The first step during preprocessing is to removethese Stop words, which has proven as very important [3].The present work uses theSMARTstop word list [4]

3.2 Stemming

Stemming techniques are used to find out the root/stem of aword. Stemming converts words to their stems, whichincorporates a great deal of language-dependent linguisticknowledge. Behind stemming, the hypothesis is that wordswith the same stem or word root mostly describe same orrelatively close concepts in text and so words can beconflated by using stems. For example, the words, user,users, used, using all can be stemmed to the word 'USE'.
In the Porter Stemmer algorithm [5], whichis the most commonly used algorithm in English, is used.
A consonant will be denoted by c, a vowel by v. A list ccc... of length greater than 0 will be denoted by C, and a list vvv... of length greater than 0 will be denoted by V. Any word, or part of a word, therefore has one of the four forms:
CVCV ... C
CVCV ... V
VCVC ... C
VCVC ... V

3.3 Document Indexing
Themain objective of document indexing is to increase theefficiency by extracting from the resulting document aselected set ofterms to be used for indexing the document.Document indexing consists of choosing the appropriateset of keywords based on the whole corpus of documents,and assigning weights to those keywords for eachparticular document, thus transforming each documentinto a vector of keyword weights. The weight normally isrelated to the frequency of occurrence of the term in thedocument and the number of documents that use that term.

3.3.1TermWeighting

In the vector space model, the documents are representedas vectors. Term weighting is an important concept whichdetermines the success or failure of the classificationsystem. Since different terms have different level ofimportance in a text, the term weight is associated withevery term as an important indicator [6].
The three main components that affect the importance of aterm in a document are the Term Frequency (TF) factor,Inverse Document Frequency (IDF) factor and Documentlength normalization [7].Term frequency of each word in adocument (TF) is a weight which depends on thedistribution of each word in documents. It expresses theimportance of the word in the document. Inverse documentfrequency of each word in the document database (IDF) isa weight which depends on the distribution of each word inthe document database. It expresses the importance of eachword in the document database [8]. TF/IDF is a techniquewhich uses both TF and IDF to determine the weight aterm. TF/IDF scheme is very popular in text classificationfield and almost all the other weighting schemes arevariants of this scheme [9]. Given a document collection'D' , a word 'w' , and an individual documentd D, the weightw_d is calculated using Equation 1.1

$w_d = f_{w,d} * \log(|D| / f_{w,D})$ (1.1)

where,
$f_{w,d}$orTF is the number of times 'w' appears in a document 'd'
D| is the size of the dataset
$f_{w,D}$ or IDF is the number of documents in which 'w' appears inD.
The result of TF/IDF is a vector with the various termsalong with their term weight. The pseudo code for thecalculation of TF/IDF is shown in following algorithm.

```
Determine TF, calculate its corresponding weight andstore
it in
Weight matrix (WM)
Determine IDF
if IDF == zero then
Remove the word from the WordList
Remove the corresponding TF from the WM
Else
Calculate TF/IDF and store normalizedTF/IDF in the
corresponding element of theweight matrix.
```

3.4 DimensionalityReduction

Document frequency (DF) is the number of documents inwhich a term occurs. DF thresholding is the simplesttechnique for vocabulary reduction. Stop word eliminationexplained previously, removes all high frequency wordsthat are irrelevant to the classification task, while DFthresholding removes infrequent words. All words thatoccur in less than 'm' documents of the text collection arenot considered as features, where 'm' is a pre-determinedthreshold. DF thresholding is based on the assumption thatinfrequent words are not informative for categoryprediction. DF thresholding easily scales to a very largecorpora and has the advantage of easy implementation. Inthe present work, during classification, the documentfrequency threshold is set as 1 so that terms that appear inonly one document are removed.

IV. CONCLUSION

The present work uses five important preprocessing techniques namely, stop word removal, stemming, document indexing and TF/IDF on Reuter's dataset. From the experimental results,it could be seen that preprocessing has a huge impact onperformances of classification. The goal of preprocessingis to reduce the number of features which was successfully met by the selected techniques. From the results it is clear that the removal of stop-words can expand words and enhance the discrimination degree between documents and can improve the system performance. TF/IDF, the most frequently used indexing technique is used to create theindex file from the resulting terms

REFERENCES

[1] K. Aas "Text categorization: A survey", Technicalreport,Norwegian Computing Center, June, 1999.

[2] Katharina, M. and Martin, S. (2004) "The Mining Mart Approach to Knowledge Discovery in Databases", NingZhong and Jiming Liu (editors), Intelligent Technologies for Information Analysis Springer, Pp. 47-65.

[3] Xue, X. and Zhou, Z. (2009),"Distributional Features for Text Categorization", IEEE Transactions on Knowledge and Data Engineering,Vol. 21, No. 3, Pp. 428-442.

[4] Salton, G. (1989), "Automatic Text Processing: TheTransformation, Analysis, and Retrieval of Information ByComputer", Pennsylvania, Addison-Wesley, Reading.

[5] Porter, M. (1980) "An algorithm for suffix stripping, Program",Vol. 14, No. 3, Pp. 130–137.

[6] Salton, G. and Buckley, C. (1988) "Term weighting approaches In automatic text retrieval, Information Processing and Management",Vol. 24, No.5, Pp. 513-523.

[7] Karbasi, S. and Boughanem, M. (2006),"Document lengthnormalization using effective level of term frequency in largecollections", Advances in Information Retrieval, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol.3936/2006, Pp.72-83.

[8] Diao, Q. and Diao, H. (2000) "Three Term Weighting and Classification Algorithms in Text Automatic Classification", The Fourth International Conference on High-Performance Computing in theAsia-Pacific Region,Vol. 2, P.629.

[9] Chisholm, E. and Kolda, T.F. (1998) "New term weighting Formulas for the vector space method in information retrieval",Technical Report, Oak Ridge National Laboratory.

[10]C. Apte, F. Damerau and S. Weiss "Towards language independent automated learning of text categorization models". Proceeding of 17th Annual ACM/SIGIR conference,1994.

[11]William W.Cohen and Yoram Singer, "Context sensitive learning methods for text categorization", In SIGIR'96: Proceeding of 19th Annual International ACM/SIGIR conference on research and development in information retrieval, 1996.

[12] R.H. Creecy, B.M. Masand, S.J. Smith and D.L. Waltz, "Trading mips and memory for knowledge Engineering", classifying census returns on the connection machine comm.. ACM, 35:48-63,1992

[13] N. Fuhr, S.Hartmanna, G. Lusting, M. Schwanter and K.Tzeras, " Rule based multistage indexing systems for large subject field", In 606-623, editor, Proceedings of RIAO'91.[14] D. Koller and M. Sahami," Toward optimal feature selection", In proceedings of the 13th international conference on machine learning 1996

[15] D.D. Lewis and M. Ringvette, "Comparison of two learning algorithm for text categorization", In Proceeding Analysis and Information Retrieval(SDAIR'94) 1994.