

Evaluating Preprocessing Techniques in Text Categorization

V. Srividhya, R. Anitha

Abstract- Now-a-days due to high availability of computing facilities, large amount of data in electronic form is generated. The data generated is to analyze so as to maximize the benefits, for intelligent decision making. Text categorization is an important and extensively studied problem in machine learning. The basic phases in text categorization include preprocessing features, extracting relevant features against the features in a database, and finally categorizing a set of documents into predefined categories. Most of the researches in text categorization are focusing more on the development of algorithms and computer techniques.

Keywords- Preprocessing, text categorization, stop word removal, stemming, TF/IDF

I. INTRODUCTION

Amazing development of Internet and digital library has triggered a lot of research areas. Text categorization is one of them. Text categorization is a process that group text documents into one or more predefined categories based on their contents [1]. It has wide applications, such as email filtering, category classification for search engines and digital libraries. Associative text classification, a task that combines the capabilities of association rule mining and classification, is performed in a series of sequential subtasks. They are the preprocessing, the association rule generation, the pruning and the actual classification. Out of these, the first step, that is, 'Preprocessing', is the most important subtask of text classification.

The importance of preprocessing is emphasized by the fact that the quantity of training data grows exponentially with the dimension of the input space. It has already been proven that the time spent on preprocessing can take from 50% up to 80% of the entire classification process [2], which clearly proves the importance of preprocessing in text classification process.

This paper discusses the various preprocessing techniques used in the present research work and analyzes the affect of preprocessing on text classification using machine learning algorithms. Section 2 gives an overview of the work in text preprocessing. Section 3 explains the preprocessing steps used. Experimental results are described in section 4. Summarization of work narrated in Section 5.

II. TEXT PREPROCESSING

The preprocessing phase of the study converts the original

textual data in a data-mining-ready structure, where the most significant text-features that serve to differentiate between text-categories are identified. It is the process of incorporating a new document into an information retrieval system. An effective preprocessor represents the document *efficiently* in terms of both *space* (for storing the document) and *time* (for processing retrieval requests) requirements and maintain good *retrieval performance* (precision and recall). This phase is the most critical and complex process that leads to the representation of each document by a select set of *index* terms. The main objective of preprocessing is to obtain the key features or key terms from online news text documents and to enhance the relevancy between word and document and the relevancy between word and category.

III . PREPROCESSING STEPS

The goal behind preprocessing is to represent each document as a feature vector, that is, to separate the text into individual words. In the proposed classifiers, the text documents are modeled as transactions. Choosing the keyword that is the feature selection process, is the main preprocessing step necessary for the indexing of documents.

This step is crucial in determining the quality of the next stage, that is, the classification stage. It is important to select the significant keywords that carry the meaning, and discard the words that do not contribute to distinguishing between the documents.

The procedure used for preprocessing the Reuter 21578 dataset is shown in Fig.1

```

for each document in Reuters 21578 do
  Remove stop-words
end for
for each remaining word in the dataset do
  Perform Stemming using Porter Stemmer and store in a vector
  (WordList)
end for
for each word in the Wordlist do
  Calculate TF/IDF and store the result in a weight matrix
end for
for each element in weight matrix
  Set the threshold 'c'
  Calculate Document Frequency (DF) for each term
  If DF < c then
  Remove the term along with its weight from weight matrix
  End if
  End for

```

Fig. 1 Preprocessing Procedure

3.1 Stop Word Removal

Many of the most frequently used words in English are useless in Information Retrieval (IR) and text mining. These words are called 'Stop words'. Stop-words, which are language-specific functional words, *are frequent words that 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 remove these Stop words, which has proven as very important [3]. The present work uses the SMART stop word list [4]

3.2 Stemming

Stemming techniques are used to find out the root/stem of a word. Stemming converts words to their stems, which incorporates a great deal of language-dependent linguistic knowledge. Behind stemming, the hypothesis is that words with the same stem or word root mostly describe same or relatively close concepts in text and so words can be conflated by using stems. For example, the words, user, users, used, using all can be stemmed to the word 'USE'. In the present work, the Porter Stemmer algorithm [5], which is the most commonly used algorithm in English, is used.

3.3 Document Indexing

The main objective of document indexing is to increase the efficiency by extracting from the resulting document a selected set of terms to be used for indexing the document. Document indexing consists of choosing the appropriate set of keywords based on the whole corpus of documents, and assigning weights to those keywords for each particular document, thus transforming each document into a vector of keyword weights. The weight normally is related to the frequency of occurrence of the term in the document and the number of documents that use that term.

3.3.1 Term Weighting

In the vector space model, the documents are represented as vectors. Term weighting is an important concept which determines the success or failure of the classification system. Since different terms have different level of importance in a text, the term weight is associated with every term as an important indicator [6].

The three main components that affect the importance of a term in a document are the Term Frequency (TF) factor, Inverse Document Frequency (IDF) factor and Document length normalization [7]. Term frequency of each word in a document (TF) is a weight which depends on the distribution of each word in documents. It expresses the importance of the word in the document. Inverse document frequency of each word in the document database (IDF) is a weight which depends on the distribution of each word in the document database. It expresses the importance of each word in the document database [8]. TF/IDF is a technique which uses both TF and IDF to determine the weight a term. TF/IDF scheme is very popular in text classification field and almost all the other weighting schemes are

variants of this scheme [9]. Given a document collection 'D', a word 'w', and an individual document $d \in D$, the weight w_d is calculated using Equation 1.1

$$[10]w_d = f_{w,d} * \log(|D|/f_{w,D}) \quad (1.1)$$

where

$f_{w,d}$ or TF 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 in D.

The result of TF/IDF is a vector with the various terms along with their term weight. The pseudo code for the calculation of TF/IDF is shown in Fig. 2.

```

Determine TF, calculate its corresponding weight and
store 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 normalized
    TF/IDF in the corresponding element of the
    weight matrix
  
```

Fig. 2 TF/IDF Algorithm

3.4 Dimensionality Reduction

Document frequency (DF) is the number of documents in which a term occurs. DF thresholding is the simplest technique for vocabulary reduction. Stop word elimination explained previously, removes all high frequency words that are irrelevant to the classification task, while DF thresholding removes infrequent words. All words that occur in less than 'm' documents of the text collection are not considered as features, where 'm' is a pre-determined threshold. DF thresholding is based on the assumption that infrequent words are not informative for category prediction. DF thresholding easily scales to a very large corpora and has the advantage of easy implementation. In the present work, during classification, the document frequency threshold is set as 1 so that terms that appear in only one document are removed.

IV. RESULTS AND DISCUSSION

Fig.3 shows the result after applying each of the preprocessing techniques, namely, stop word removal, stemming and TF/IDF when combined with Document Frequency Thresholding method, while varying the threshold value. The experiment was conducted with 20 documents, with a total number of approximately 2,15,000 unique term. It was decided to perform Stop Word Removal (SWR) and DF Thresholding (DFT) compulsorily. The reason behind such a decision was that, SWR and DFT, when omitted, will result in a vector space

having unimportant features. The experiments were conducted using seven document frequency threshold values, namely, 50, 100, 200, 400, 800, 1200 and 1600. The thresholding is the DF value rather than the percentage. These threshold values were chosen in a random fashion. Thus the experiments conducted had the following setups.

- (i) SWR + DFT
- (ii) SWR + DFT + TF/IDF
- (iii) SWR + DFT + Stemming + TF/IDF

From the Figure, it could be seen that the application of all the preprocessing techniques have a positive impact on the number of terms selected. The results further reveal an important fact that stemming, even though is very important is not making only very negligible difference in terms of number of terms selected.

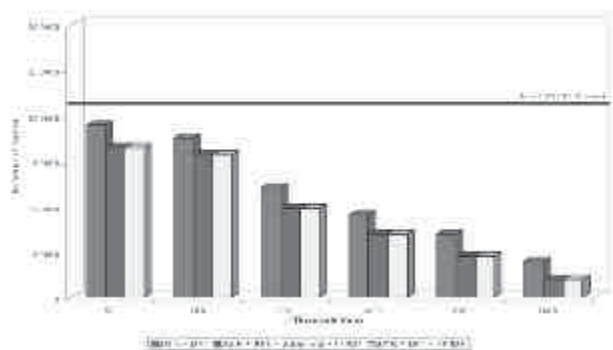


Fig. 3 Effect of Preprocessing

V. CONCLUSION

The present work uses three important preprocessing techniques namely, stop word removal, stemming and TF/IDF on Reuters dataset. From the experimental results, it could be seen that preprocessing has a huge impact on performances of classification. The goal of preprocessing is 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 the index file from the resulting terms.

REFERENCES

- [1] K.Aas and A.Eikvil, "Text categorization: A survey", Technical report, Norwegian Computing Center, June, 1999.
- [2] Katharina, M. and Martin, S. (2004) The Mining Mart Approach to Knowledge Discovery in Databases, Ning Zhong 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: The Transformation, Analysis, and Retrieval of Information by Computer*. 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 length normalization using effective level of term frequency in large collections, *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 the Asia-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] Berger, A., Caruana, R., Cohn, D., Freitag, D. and Mittal, V. (2000). Bridging the Lexical Chasm: Statistical Approaches to Answer Finding, *Annual ACM Conference on Research and Development in Information Retrieval archive, Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Publishers, Pp. 192-199.



V. Srividhya

Assistant Professor,
Avinashilingam University for Women,
Coimbatore.
vidhyavas@gmail.com