Evaluation of Pre Processing Techniques for Information Storage and Retrieval System

[1]Dr. Dharmendra Sharma, [2]Dr. Suresh Jain, [3]Dr. Harish Nagar

[1]Associate Professor, [2]Dean of Academics, [3]Research Head
[1] Department of Computer Engineering
[1]NMIMS Indore, India

Abstract: ***Enormous amount of information is available over the internet in electronic format but collecting and cataloging the relevant information is very challenging task. The large amount of data can be analyzed to optimize the benefits, for business intelligent system. The storage and retrieval of information is an important and extensively studied problem in machine learning. The basic steps in information retrieval include preprocessing of document, extracting relevant terms against the features in a database, and finally categorizing a set of documents into predefined categories. In this paper we are had use two preprocessing activity as stop word removal and stemming and evaluated the impact of stop word and stemming onto feature selection. From the results we find that the removal of stop-words decrease the size of feature set. For sparsity value 0.9 it decrease by 9%.On the other hand for stemming process as we increase the sparsity value the size of feature set also increase.***

Index Terms – Information Retrieval, Term, and learning, stemming, feature selection

---

I. INTRODUCTION

Amazing development of Internet and digital library has triggered a lot of research areas. Information Retrieval is one of them. Information Retrieval 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 categorization for search engines and digital libraries. Associative text categorization, a task that combines the capabilities of association rule mining and categorization, is performed in a series of sequential subtasks. They are the preprocessing, the association rule generation, the pruning and the actual categorization. Out of these, the first step, that is, 'Preprocessing', is the most important subtask of text categorization. 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 from50% up to 80% of the entire categorization process [2], which clearly proves the importance of preprocessing in text categorization process. This paper discusses the various preprocessing techniques used in the present research work and analyzes the affect of preprocessing on text categorization 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. RELATED WORK

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. Our method is the evaluation of the weighting methods. Until now, there are many researches about weighting method. The reference [3] describes survey about the weighting method s such as binary, term frequency (TF), augmented normalized term frequency [4], log, inverse document frequency (IDF) [5].

III. METHODOLOGY

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 categorization 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 web document dataset is shown in Fig.1

Step 1:Data Collection
Step 2:Stop word removal
Step 3: Stemming
Step 4: Indexing
Step 5: Term weighting
Step 6:Feature Selection

Fig. 1: Processing steps

3.1. Data Collection

For the data set we use Google web API to collect the document. We have collected 2064 documents related to jio users it has 209998 features.

3.2 Stop Word Removal

Stop word are functional words that do not carry information. 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 [6]. Most of the frequently used words in English sentence are useless in Information Retrieval (IR) and text mining.

3.3 Stemming

Stemming is a techniques used to find out the root or stem of a word. Stemming converts words to their root words, 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, which is the most commonly used algorithm in English, is used [7].

3.4 Document Indexing

The objective of indexing is to increase the efficiency of the system by selecting the appropriate term from 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.

3.5 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 Information Retrieval system. Since different terms have different level of importance in a text, the term weight is associated with every term as an important indicator [8].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 [9].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 [10]. TF/IDF is a technique which uses both TF and IDF to determine the weight of term. TF/IDF scheme is very popular in text categorization field and almost all the other weighting schemes are variants of this scheme [9]. In vector space model organization of document also affect the performance of system [11].In this experiment we use term frequency method, other are also acceptable.

3.6 Feature Selection

The document term matrix contains the set of document as row and set of terms as columns. These terms are also known as features because there are used to uniquely identify the document. The sparsity of document term matrix represent the set of features that's frequency is zero. Higher the sparsity value lead to increase the set of feature and lower the sparsity value decrease the set of feature. Document frequency (DF) is the number of documents in which a term occurs. DF thresholding is the simplest technique for feature reduction. Stop word elimination explained previously, removes all high frequency words that are irrelevant to the categorization 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 categorization, the document frequency threshold is set as sparsity of matrix that varies from 0.1 to 0.9.

4 SYSTEM SETUP

We have used Google Web API for aggregation of web data. We have aggregate 2064 contents for concept "Jio user". This experiment system is implemented by R including tm package. We have created one matrix as a document term matrix that has 209998 features.

5 RESULT AND DISCUSSION

The experiment was conducted with 2064 documents, having 209998 unique terms. The experiments have been conducted using nine documents frequency threshold values (sparsity value in %), namely, 10, 20, 30, 40, 50, 60, 70, 80 and 90. The thresholding is the percentage value rather than the sparsity value. Table 1 and 2 show the result after applying the preprocessing technique namely stop word removal and stemming

TABLE 1: IMPACT OF STOP WORD WITH DIFFERENT SPARSITY VALUE ON FEATURE SET

Sparsity Value	With Stop Word	Without stop word
0.1	19	11
0.2	117	14
0.3	118	16
0.4	139	132
0.5	171	146
0.6	1141	189
0.7	1215	1070
0.8	1442	1374
0.9	11019	1936
1.0	19998	19793

From the table 1 it is observed that the elimination of stop-words decrease the size of feature set. We found the maximum decrement in feature set at sparsity value 0.9 as 90%.

TABLE 2: IMPACT OF STEMMING WITH DIFFERENT SPARSITY VALUE ON FEATURE SET

Sparsity	Without Stemming	With Stemming
0.1	11	10
0.2	11	12
0.3	16	19
0.4	122	131
0.5	146	159
0.6	189	1115
0.7	1170	1217
0.8	1374	1411
0.9	1936	1930
1.0	19793	17338

Table 2 shows the impact of stemming on feature set for different sparsity value. From the table 2 it is clear that the stemming process affect significantly to the size of feature set with different sparsity value. As we increase the sparsity value the size of feature set also increase. Only for sparsity value 0.9 the feature set decrease from 19793 to 1936. From table 1 and 2, it could be seen that the application of stop word removal and stemming 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.

6 CONCLUSION

The presented work uses two important preprocessing techniques namely, stop word removal and stemming on web dataset. From the experimental results, it could be seen that preprocessing has a huge impact on performances of categorization. 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 decrease the size of feature set. For sparsity value 0.9 it decrease by 9%.On the other hand for stemming process as we increase the sparsity value the size of feature set also increase. Only for sparsity value 0.9 the feature set decrease from 19793 to 1936.

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] T.G.Kolda ,D.P.O'Leary,"A semi discrete matrix decomposition for latent semantic indexing information retrieval", Journal ACM Transactions on Information Systems (TOIS)TOISH one page archive vol.16(4), pp. 322-346, Oct. 1998.

[4]G.Salton, C.Buckley,"Term weighting approaches in automatic text retrieval," *Inf. Process. Manage*. 24, pp. 513–523, 1988.

[5] D.Harman, "Ranking algorithms. In Information Retrieval: Data Structures and Algorithms, "*W.B.Frakesand R. Baeza-Yates, Eds. Prentice Hall, Englewood Cliffs*, NJ, pp.363–392, 1992.

[6] 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.

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

[8] 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.

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

[10] 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.

[11] Sharma Dharmendra, jain suresh, “Content sharing in information storage and retrieval system using tree representation of documents”,IEEE ,International conference on IT industry, business and government,CSIBIG2014 page 1-4,2014