Automatic Document Collection

Shashikant
M. Tech, Faculty Of Electronics,
CS & Informatics, Shobhit University, Meerut.
8/2434, Matagarh,
Saharanpur, U.P.

Mukesh Rawat
Faculty Of Electronics, CS & Informatics, Shobhit University, Meerut.
C.S.E. Department, Shobhit University, Meerut, U.P.

ABSTRACT

Now a day's classification of document is an important area for research, as large amount of electronic documents are available in form of unstructured, semi structured and structured information. Document classification will be applicable for World Wide Web, electronic book sites, online forums, electronic mails, online blogs, digital libraries and online government repositories. So it is necessary to organize the information and proper categorization and knowledge discovery is also important. This paper focused on the existing literature and explored the techniques for automatic documents classification i.e. documents representation, knowledge extraction and classification. In this paper author propose an algorithm and architecture for automatic document collection.

General Terms

Document classification, Pattern Recognition, Classification algorithms.

Keywords

Text mining, Web mining, Automatic document classification.

1. INTRODUCTION

Documents classification studies are gaining more importance recently because of the availability of the increasing number of the electronic documents from a variety of sources. The resources of unstructured and semi structured information include the world wide web, governmental electronic repositories, news articles, biological databases, chat rooms, digital libraries, online forums, electronic mail, and blogs repositories. So extracting information from these resources and proper categorization and knowledge discovery is an important area for research.

Text mining deals the categories of operations, retrieval, classification (supervised, unsupervised and semi supervised) summarization, trend and association analysis. The main goal of text mining is to enable users to extract information from textual resources. How the documented can be proper annotated, presented and classified, so the documents categorization consist several challenges, proper annotation to the documents, appropriate document representation, an appropriate classifier function to obtain good generalization.

Today web is the main resource for the text documents. The amount of textual data available to use is consistently increasing; approximately 80% of the information of an organization is stored in unstructured textual form in the form of reports, email, views and news. Information intensive business processes demand that user transcend from simple document retrieval to "knowledge" discovery. The need of automatically extraction of useful knowledge from the huge amount of textual data in order to assist the human analysis is fully apparent. Market Trends based on the content of the online news articles, sentiments, and events is an emerging topic for research in data mining and text mining community.

2. LITERATURE REVIEW

A wide variety of techniques have been designed for text classification. In this chapter, author will discuss the broad classes of techniques, and their uses for classification tasks. It is noticed that these classes of techniques also generally exist for other data domains such as quantitative or categorical data. Since text may be modeled as quantitative data with frequencies on the word attributes, it is possible to use most of the methods for quantitative data directly on text. However, text is a particular kind of data in which the word attributes are sparse, and high dimensional, with low frequencies on most of the words. Therefore, it is critical to design classification methods which effectively account for these characteristics of text. In this chapter, focus on the specific changes which are applicable to the text domain. Some key methods, which are commonly used for text classification, are as follows:

2.1 Decision Trees

Decision trees [3] are designed with the use of a hierarchical division of the underlying data space with the use of different text features. The hierarchical division of the data space is designed in order to create class partitions which are more skewed in terms of their class distribution. For a given text instance, we determine the partition that it is most likely to belong to, and use it for the purposes of classification.

2.2 Pattern (Rule)-based Classifiers

In rule-based classifiers [4] we determine the word patterns which are most likely to be related to the different classes. We construct a set of rules, in which the left-hand side corresponds to a word pattern, and the right-hand side corresponds to a class label. These rules are used for the purposes of classification.

2.3 SVM Classifiers

SVM Classifiers [5] attempt to partition the data space with the use of linear or non-linear delineations between the different classes. The key in such classifiers is to determine the optimal boundaries between the different classes and use them for the purposes of classification.

2.4 Neural Network Classifiers

Neural networks are used in a wide variety of domains for the purposes of classification. In the context of text data, the main difference for neural network classifiers is to adapt these classifiers with the use of word features. It is noticed that neural network classifiers are related to SVM classifiers;

indeed, they both are in the category of discriminative classifiers, which are in contrast with the generative classifiers [9].

2.5 Bayesian (Generative) Classifiers

In Bayesian classifiers [6] (also called generative classifiers), it attempts to build a probabilistic classifier based on modeling the underlying word features in different classes. The idea is then to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word presence in the documents.

2.6 Other Classifiers

Almost all classifiers can be adapted to the case of text data. Some of the other classifiers include nearest neighbor classifiers, and genetic algorithm-based classifiers.

3. PROPOSED WORK

In this proposed methodology, HTML documents are used as an input and converted into text file. Some lines are selected from the text file and forwarded to the Tokenizer. That Tokenizer removes the stop-words from these lines. These tokens again forwarded to word matcher that matches the words with the index files. If 40% and above tokens matched with one index file then document is belongs to concern index topic and get moved to concern topic repository.

In this way, the document classification has been done according to content of HTML pages.

3.1 Used Tools and Platform

Platform used in this methodology is Java.

IDE used - Net bean 7.2, IDE for the implementation of this approach.

3.2 Modules used in proposed methodology

In this section, describing the following modules in brief which are used in proposed architecture. They are-

3.2.1 Convertor & Parser

This module is used to convert the HTML documents to text file. The parser will remove the HTML tags from the text file and select the starting eight lines and also the middle eight lines from the same text file.

Proposed Algorithm for parser – which convert html page to text and select the lines for token generation

```
Parser ()
{
  If(HTML page)
  {
    Txt file  ← HTML file;
  }
   If(txt File)
   {
     While(EOF)
       {
         Text file ← Remove HTML tags;
       }
     }
 If (Text file)
{
   While (EOF)
   {
      Line_no ++;
    }
  Mid_no = (1+line_no)/2;
  While(temp<8)
   {
     String1 ← Lines;
    }
     Temp=mid_no;
   While(temp<temp+8)
    {
      String2←Lines;
    }
   FString=String1+String2;
}
Return(Fstring)
}
```

3.2.2 Tokenizer

This module will remove the stop-words from the lines. After removing stop-words it generates the tokens.

3.2.3 Word Matcher

This is the main module of proposed architecture. It matches the tokens with the indexes available. If the matched tokens equal to or greater than 40% of index words then it belongs to concern index repository.

Proposed algorithm for Word-matcher()

```
Word_matcher()
{
  If (token)
  {
    While(token == word dictionary)
     {
        calculate match token percentage;
     }
   }
```

```
    If (match>= 40%)
    {
        Store document in specific domain cluster;
     }
      Else
      {
        Parse whole document to generate token;
      }
}
```

3.2.4 Parse whole file

If the matched tokens are less than 40% of index words then the whole text file will be recycled. And send to parser then tokens will be generated from the whole text file.

3.2.5 Index

In this module indexes are available for different topics, which get matched with tokens.

3.2.6 Domain specific clustering

When matching phase is completed then 40% matched or greater matched document moved to concern specified domain name repository.

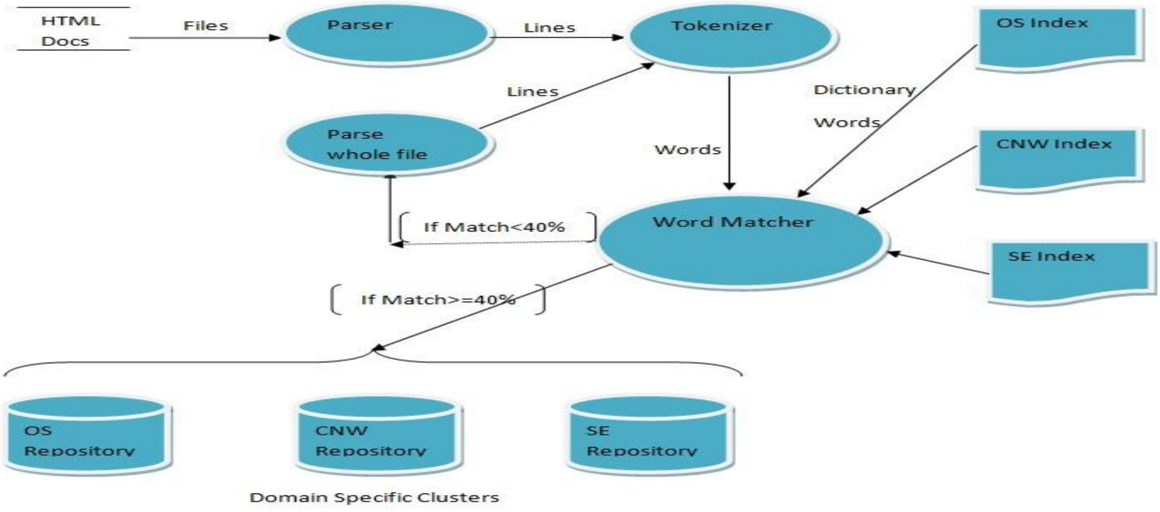
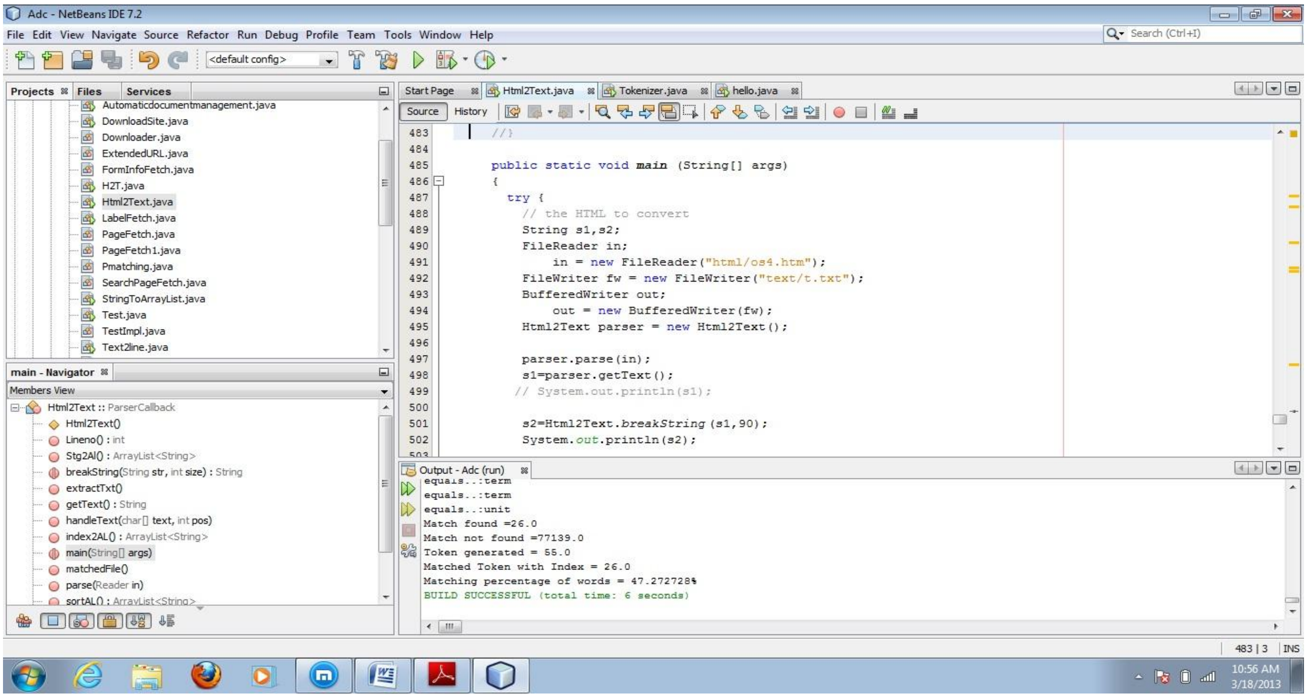


Figure 1: Proposed Architectutre For Automatic Document Collection



```
        //}

        public static void main (String[] args)
        {
          try {
            // the HTML to convert
            String s1,s2;
            FileReader in;
                in = new FileReader("html/os4.htm");
            FileWriter fw = new FileWriter("text/t.txt");
            BufferedWriter out;
                out = new BufferedWriter(fw);
            Html2Text parser = new Html2Text();

            parser.parse(in);
            s1=parser.getText();
           // System.out.println(s1);

            s2=Html2Text.breakString(s1,90);
            System.out.println(s2);
```

```
equals..:term
equals..:term
equals..:unit
Match found =26.0
Match not found =77139.0
Token generated = 55.0
Matched Token with Index = 26.0
Matching percentage of words = 47.272728%
BUILD SUCCESSFUL (total time: 6 seconds)
```

Figure 2: Netbean IDE 7.2

4. CONCLUSION AND FUTURE SCOPE

In this paper, author has presented an approach that works as self organization of documents. It classifies the HTML documents on content based technique. It extracts the paragraphs from web pages and then generates the tokens. Our preliminary experimental results demonstrate that classification on information extraction based technique working well.

Currently this technique is working on few domains specific; by parallel processing it can be implemented on many domains simultaneously. It can be implemented on hierarchal based classification, firstly main topic classification then sub topic classification. It can improve its efficiency.

5. REFERENCES

[1]. Aurangzeb Khan, Baharum B. Bahurdin, Khairullah Khan, “An Overview of E-Documents Classification”, 2009 International Conference on Machine Learning and Computing IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore.

[2]. S. Gopal, Y. Yang. Multilabel classification with meta-level features. ACM SIGIR Conference, 2010.

[3]. J. R. Quinlan, Induction of Decision Trees, Machine Learning,1(1), pp 81–106, 1986.

[4]. B. Liu, W. Hsu, Y. Ma. Integrating Classification and Association Rule Mining. *ACM KDD Conference*, 1998.

[5]. C. Cortes, V. Vapnik. Support-vector networks. Machine Learning, 20: pp. 273–297, 1995.

[6]. M. Sahami. Learning limited dependence Bayesian classifiers, *ACM KDD Conference*, 1996.

[7]. B. Liu, L. Zhang. A Survey of Opinion Mining and Sentiment Analysis. Book Chapter in Mining Text Data, Ed. C. Aggarwal, C. Zhai, Springer, 2011.

[8]. M. Sahami, S. Dumais, D. Heckerman, E. Horvitz. A Bayesian approach to filtering junk e-mail. AAAI Workshop on Learning for Text Categorization. Tech. Rep. WS-98-05, AAAI Press. http://robotics.stanford.edu/users/sahami/papers.html.

[9]. A. Y. Ng, M. I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. NIPS. pp. 841- 848, 2001.

[10]. J. R. Quinlan, Induction of Decision Trees, Machine Learning, 1(1), pp 81–106, 1986.

[11]. P. Long, R. Servedio. Random Classification Noise defeats all Convex Potential Boosters. ICML Conference, 2008.

[12]. S. A. Macskassy, F. Provost. Classification in Networked Data: AToolkit and a Univariate Case Study, Journal of Machine Learning Research, Vol. 8, pp. 935–983,2007.