International Journal of Scientific Research in Computer Science and Engineering
Research Paper Volume-2, Issue-5 ISSN: 2320-7639

Web Text Content Extraction and Classification using Naïve Bayes Classifier Algorithm

Sanjay S Bhadoria[1*] and Rajendra Kumar Patel[2]

[1*,2]*Department of Computer Science, PCST, Bhopal*
sanjay.bhadoria@gmail.com and sahilsinghmanit@gmail.com

Available online at www.isroset.org

Received: 10 Sep 2014 Revised: 22 Sep 2014 Accepted: 12 Oct 2014 Published: 31 Oct 2014

Abstract - The Web today contains lots of information about subjects such as people, companies, organizations, products, etc. That may be of wide interest. Text mining is the technique that helps users to find useful information from a large amount of digital text documents on the Web or databases. This paper discusses The naive Bayes classifier algorithm of how to follow the appointed website or web page according to users request and in Internet by extraction on web mining.

Keywords- Classification, Text Extraction, Link Crawler, Data Mining

I. INTRODUCTION

Today the Web has become the largest information source for people. Copious material is available from the World Wide Web (www) in response to any user-provided query. Large document collections, such as those delivered by Internet search engines, are difficult and time-consuming for users to read and analyze. It becomes tedious for the user to manually extract real required information from this material. Though the physical characteristics of Web information is distributed and decentralized, the WWW can be viewed as one big virtual document collection. In that regard, the fundamental questions and approaches of traditional Information Retrieval (IR) research (e.g.term weighting, query expansion) are likely to be relevant in Web document retrieval. The Web document collection, massive in size and diverse in content, context, format, purpose and quality, challenges the validity of previous research findings based on relatively small and homogeneous test collections. Keeping information organized is an important issue to make information access easier. Although the information we need is sometimes available on the Web, this information is only useful if we have the ability to find it. With this aim, it is increasingly frequent to use automatic techniques for grouping documents.

With more than two billion pages created by millions of Web page authors and organizations, the World Wide Web is a tremendously rich knowledge base[01]. The knowledge comes not only from the content of the pages themselves, but also from the unique characteristics of the Web, such as its hyperlink structure and its diversity of content and languages. A considerably large portion of information present on the World Wide Web (www) today is in the form of unstructured or semi-structured text data bases. Figure 1 depicts a generic process model for a text mining application. Starting with a collection of documents, a text mining tool would retrieve a particular document and preprocess it by checking format and character sets. Then it would go through a text analysis phase, sometimes repeating techniques until information is extracted. The resulting information can be placed in a management information system, yielding an abundant amount of knowledge for the user of that system [4].

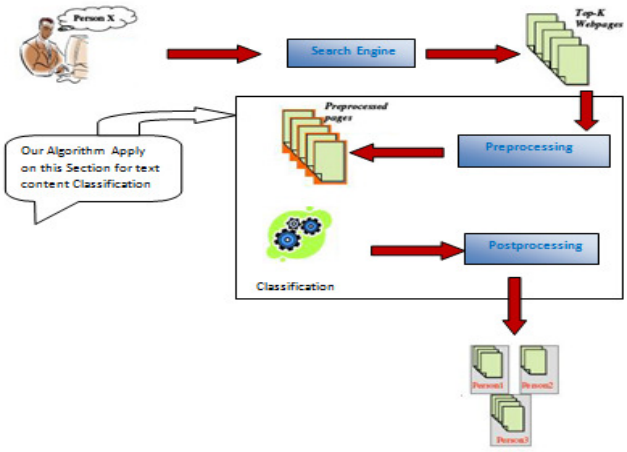

Fig 1: depicts a generic process model for a text mining application

II. RELATED WORKS

Website classification has been studied for quite a long time and much effort has been made to improve and develop new methods to obtain better results. Tsukada *et al.* (2001) put forward a method to use a Japanese thesaurus for the classification of Japanese websites [2]. Their average recall, precision and error rates were 48.0%, 74.6% and 13.7% respectively. The thesaurus they used did not generate satisfactory performance. Choi and Peng (2004) suggested an approach that makes use of class hierarchies for improving web page classification [3]. They developed a single path

Corresponding Author: *S Bhadoria*

algorithm to sort the websites in a classification tree. They obtained a classification accuracy of 84.5% when tested on 339 websites. However, they concentrated on only one category and hyperlinks were not considered.

III. Scope of the work

In this paper, a Naïve Bayes algorithm for web text mining is proposed. This algorithm gives better execution time as compare too many previous algorithms like Apriori, SVM, K-nearest Neighbor and Decision Making Tree. It also supports the incremental nature of database.

Our proposed algorithm crawling a web pages from any URL display all crawled links of pages. We have selected any link of them and extract all the content from the page and uses for classification. We also store all data in data base for future uses. For extracting data, particular text (news) is extracted from the desired webpage. Text extraction is done with the help of html parser. After text extraction preprocessing of the text is done. Preprocessing is removing irrelevant words for classifying data and each preprocessed word is checked with the stored collection of dataset. Dataset contain 10 categories, depending on the words in the text, text is categorized in particular category from all 10 categories. Information having better match of words with particular category is classified in that particular category.

System can also perform and multilevel sub classification:

- In news can be further classified in two categories depending on words match between news and category. Example any sports news at national level, suppose cricket news may contain words matching with sports category as well as nation category.
- In sub classification each of 10 categories can be further classified in two sub categories. For example news is classified in sports, education, business and national. Each category is classified further such as education into school and college.

IV. EXPERIMENTAL SETUP

Our choice of categories was based on the availability of an adequate number of websites based on these categories. We have therefore defined seven broad categories namely “Movie”, “Study”, “Government”, “Flower”, “Education”, “Technology”, “Jobs”, ” Games”, ” News ”, ” Science ”, and” Sports ”. A hard categorization approach was adopted in this study. This means that a website can be classified in not more than one category. We have opted to use a keyword-based algorithm so the need to get the right collection of keywords relevant to each category was of prime importance. This work will done in two phase. In 3 phase

A. Crawl the URL
B. Extract the text content
C. Prepossessing
D. Text Computation

A. Crawl the URL

We have crawled the all links of given URL and display on browser and store in database. A Web crawler starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites it copies and saves the information as it goes.

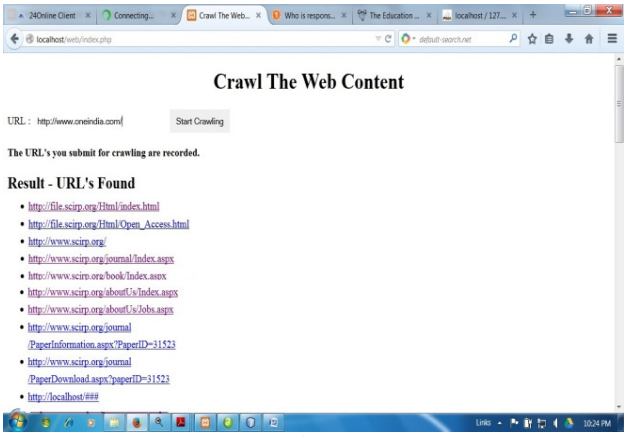

Fig 2

Select any URL from crawled links for text extraction. We have also store all URL in database for future uses.

B. Extraction the text content

Extraction of required text from web page is one of the important parts of project. As the web pages on any website contains ample of unimportant data from user‘s point of view. From large chunks of HTML code, without knowing its structure or the tags used, extracting the required text is the main task. Extracting text from arbitrary HTML files doesn‘t necessarily require scraping the file with custom code. As all the web pages are designed by html and xml, html parser can be used for parsing the text and create text file and store all content in text file.

C. Prepossessing

Preprocessing is prior step of classification. It is done with the help of stemming and stop word removal. Word Stemming is common form of language processing in most Information Retrieval (IR) systems. Word stemming is an important feature supported by present day indexing and search systems. Basic idea is to improve recall by automatic handling of word endings by reducing the words to their word roots, at the time of indexing and searching. Stemming is usually done by removing any attached suffixes, and prefixes from index terms before the assignment of the term. Since the stem of a term represents a broader concept than the original term, the stemming process eventually increases the free text-searching, searches exactly as it is typed in to the search box, without changing it to thesaurus term. It is difficult for the end user to decide upon which all terms to key in and get the results. At this point word stemming will be needed. It is observed that in most cases, morphological variants of words

have similar semantic interpretations. By stemming dictionary size is reduced. A smaller dictionary size results in a saving of storage space and processing time.

We create database for storing stemming word after remove all stop word. This data set is used for search the classification of text. Word database shown in fig 4.

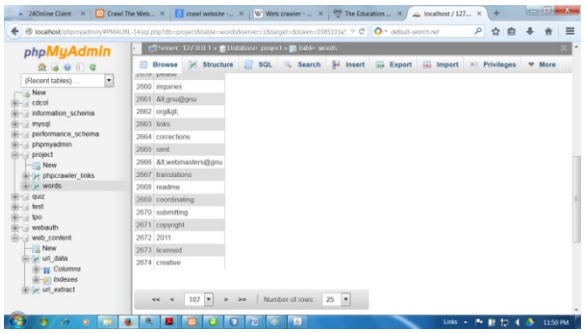

Fig 3

D. Text Computation.

To classify each website content in text file, we have to assign a name to each category. The score of each category is the sum of the matches obtained by the array of keywords, provided in text file, belonging to that particular category. Moreover, the location of the matches is also taken into consideration where different weights are applied to each location. For example if the keyword "school" which belongs to the "education" text file is found in the text contents of a webpage. This work is done in JAVA scrip by Naïve Bayes algorithm.

After run the java application the algorithm count the frequency and score the weight of the word and after assign the category in which file count highest score.

Output of Algorithm:-

Run:

category:Movie

category:Study

category:Flower

category:Technology

category:Jobs

category:Research Publication

category:Games

category:News

category:Science

category:Sports

knowledgeBase.n:23

knowledgeBase.c:10

featureOccurrencesInCategory:382.0

featureOccurrencesInCategory:2.0

featureOccurrencesInCategory:129.0

featureOccurrencesInCategory:354.0

featureOccurrencesInCategory:129.0

featureOccurrencesInCategory:97.0

featureOccurrencesInCategory:218.0

featureOccurrencesInCategory:2.0

featureOccurrencesInCategory:58.0

featureOccurrencesInCategory:296.0

in the above output the calculate occurrences of the word in each category.

featureCategory:prepare

featureCategoryCounts:{Research Publication=1}

featureCategory:functions

featureCategoryCounts:{Movie=1}

featureCategory:chips

featureCategoryCounts:{Games=1}

featureCategory:numerous

featureCategoryCounts:{Sports=1}

featureCategory:exhibit

featureCategoryCounts:{Sports=1}

featureCategory:publishing

featureCategoryCounts:{Research Publication=1}

featureCategory:chopping

featureCategoryCounts:{Games=1}

featureCategory:they

featureCategoryCounts:{Movie=1, Research Publication=1, Games=2, Sports=1}

featureCategory:vibrant

featureCategoryCounts:{Technology=1}

featureCategory:opponents

featureCategoryCounts:{Sports=1}

featureCategory:ancient

featureCategoryCounts:{Movie=1}

featureCategory:accept

featureCategoryCounts:{Research Publication=1}

featureCategory:actors

featureCategoryCounts:{Movie=1}

featureCategory:124

In the above output calculate weight of each word in all file. We search any word category by enter in text field given in fig 5.

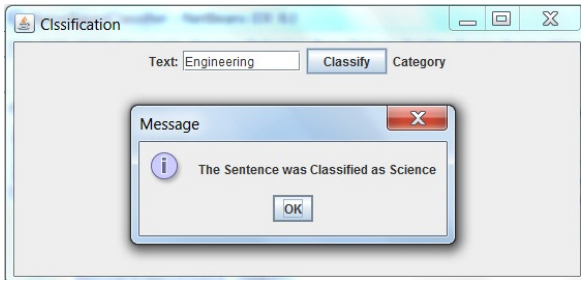

Fig. 4

We have crawled 1815 URL shown in fig 4 and extract content from 277 URL from the crawled list of URL shown in fig 7. Then we create dataset of 2670 word shown in fig 8. We have better perform the result in 2670 word for classification of any query.

	500 Key words	1500 Keywords	2670 Key words
Correctly Classified websites	159 (57.4%)	241 (87%)	260 (93.86%)
Incorrectly Classified Websites	118 (42.6%)	36 (13%)	7 (6.14%)

Tab 1 Accuracy using different number of keywords

The accuracy of 93.86% was obtained by using 260 keywords. It means if the no of word store in database then the accuracy of searching in defiantly increase.

Three performance measures were used to evaluate the system namely precision, recall and the F-measure. Precision is the number of websites correctly classified divided by the number of websites classified in that category. Recall is the number of websites correctly classified divided by the number of websites that should have been classified in a particular category. F-measure is a weighted average based on recall and precision. It provides a single measure of accuracy

Category	Precision	Recall	F-Measure
Movie	98.2	97.6	99.1
Study	88	91	96
Flower	94	92	96
Technology	91.5	96.4	97.2
Jobs	94	94.6	99
Research Publication	100	99	99.7
Games	89	94	95
News	99	98	97.9
Science	94.6	96.7	99
Sport	89	91	93.6

Tab 2 Classification Accuracy

Tab 2 shows an average precision of 93.7%, an average recall of 95.03% and an average F-measure of 97.25%. We have obtained a precision of 100% in the research publication

V. Conclusion

In this research we set out to build classification aware web content detection systems. We can develop different models for automatically classifying web pages into pre-defined topic categories in Future. Word weighting, text classification and sentiment analysis techniques are applied to extract topic. We have used Naïve Bayes an efficient and effective algorithm whereby each website was given a score for each category and the scores were then compared to classify the websites. Our system yielded an overall accuracy of 96%. With the huge number of websites, search engines needs to produce results which are both highly relevant and precise.

Reference

[01] Shaun Yin Gang Wang Yaqui Qiu Weiqun Zhang. ‖ Research and Implement of Classification Algorithm on Web Text Mining‖. IEEE.(2007)446-449

[02] Choi, B. and Peng, X., 2004. Dynamic and Hierarchical Classification of Web Pages. Online Information Review, Vol. 28, No. 2, pp. 139-147.

[03] Sam, L. Z., Maarof, M. A. B. and Selamat, A., 2006. Automated Web Pages Classification with Independent Component Analysis. Proceedings of the Postgraduate Annual Research Seminar. Vol. 1, pp. 466-469.

[03]. M. Castellano, G. Mastronardi, A. Aprile, and G. Tarricone ‖A Web Text Mining Flexible Architecture‖. World Academy of Science, Engineering and Technology 32 2007

[04] Catarina Silva, Bernardete Ribeiro ―Margin-based Active Learning and Background Knowledge in Text Mining‖.Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS'04)IEEE

[05] Weiguo Fan1, Linda Wallace, Stephanie Rich, Zhongju Zhang ―Tapping into the Power of Text Mining‖.

[06] http://tartarus.org/~martin/PorterStemmer

[08] http://www.htmlparser.com

[09] Mahadevan, I., Karuppasamy, S. and Ramasamy, R., 2009. Resource Optimization in Automatic Web Page Classification using Integrated Feature Selection and Machine Learning. International Arab Journal of e-Technology, Vol. 1, No. 1, pp. 19-28.

[10] Zhang, B., Xu, M. and Xiu, L., 2012. A Web Site Classification Approach Based on its Topological Structure. International Journal on Asian Language Processing. Vol. 20, No. 2, pp. 75-86.

