Realization of Framework for Web Content Extraction and Classification

Ganesh D. Puri
M.Tech Information Technology,
Bharti Vidyapeeth University COE,
Pune-46, Maharashtra, India.

Prof. Y.C. Kulkarni
Assistant Professor, Information Technology, Bharti Vidyapeeth University COE,
Pune-46, Maharashtra, India.

ABSTRACT
Web content extraction and classification can be viewed as combination of different methods. Nowadays web page contains lot of information including main contents. Contents extraction which are of user's interest is main task. 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. It is therefore crucial that a good text mining model should retrieve the information that meets user's needs within a relatively efficient time frame. A first step toward any Web-based text mining effort would be to collect a significant number of Web mentions of a subject. Thus, the challenge becomes not only to find all the subject occurrences, but also to filter out just those that have the desired meaning. The system described in this paper is capable of extracting main content and classify it. Vector space model method is used for classification.

Keywords
MVC Architecture, VSM model, Text Mining ,Extraction, Classification.

1. INTRODUCTION
Nowadays web has become large source of information. People usually use the search engine—Google, Yahoo etc. to browse the Web information mainly. Content extraction can be done by combining different extraction algorithms. These algorithms can be combined in serial fashion. In this serial method output of first algorithm can be given as input to the next algorithm until we get main content. The content extraction algorithms can be combined parallel fashion also. In this approach input is same for all algorithms .The outputs of different algorithms can be given to union or intersection depending on methods used.[5] Afterwards the contents can be given to the classification module.

2. MVC ARCHITECTURE
The aim of MVC architecture is to separate the business logic and data of the application from the presentation of data to the user. Following is the small description of each of the components in MVC architecture.

2.1 Model
It represents the data of application. The changes can be made to data by the methods. In strut Plain Old Java object handle changes. The changes will be replicated to the presentation part. The way of accessing data is defined in the Model. It just provides service to access the data and modify it.

2.2 View
This is very important part as it represents the presentation. It takes the content from the model and use it for presentation. The model behavior is defined by view. The view is not dependent on data or application logic changes and remains same even if the business logic undergoes modification. This part is represented by the Java server pages in strut technology.

2.3 Controller
The requests of user are performed by controller. It takes the requests from view and handles it to model for appropriate action. Based on the result of the action on data, the controller directs the user to the subsequent view.

3. WEB TEXT MINING
The framework shown in the Fig. 1 can be implemented using MVC architecture. Web text mining process and working of Web text mining process is explained in short.[2].

•In 1st step Lookup resources gets data from the target web document.

•In 2nd step Information extraction responsibility is to carry preprocessing and extracting the required text from web document. For example automatically clean advertisement conjunction.

•In 3rd step Mode discovery includes technique for classification and gives output to analysis stage.

•In 4th step analysis verifies and explains the mode produced in step 3. i.e. classified data is given.

The user can give the data to generate the webpage. It is done by taking URL of webpage from user. The webpage is generated from remote site. This webpage is input to the HTML parser which removes unwanted text and images.[3] The parser is acting as filter for system. The result of parser is tag tree representation of main content. Then the result is given to different procedures in parallel fashion.[9] In the first procedure it removes the prepositions and starting words like a, an, the etc. words from main content. In the second procedure stop words and quotation marks are removed. The result of the procedures can be intersected to get pure contents. The result of intersection is given to porter algorithm. It is useful to remove suffixes and prefixes from the content to get the pure words. Vector space model method is used to do the classification. It is done by voting approach.[9] After the classification the data can be used for the construction of web page again through the HTML parser.

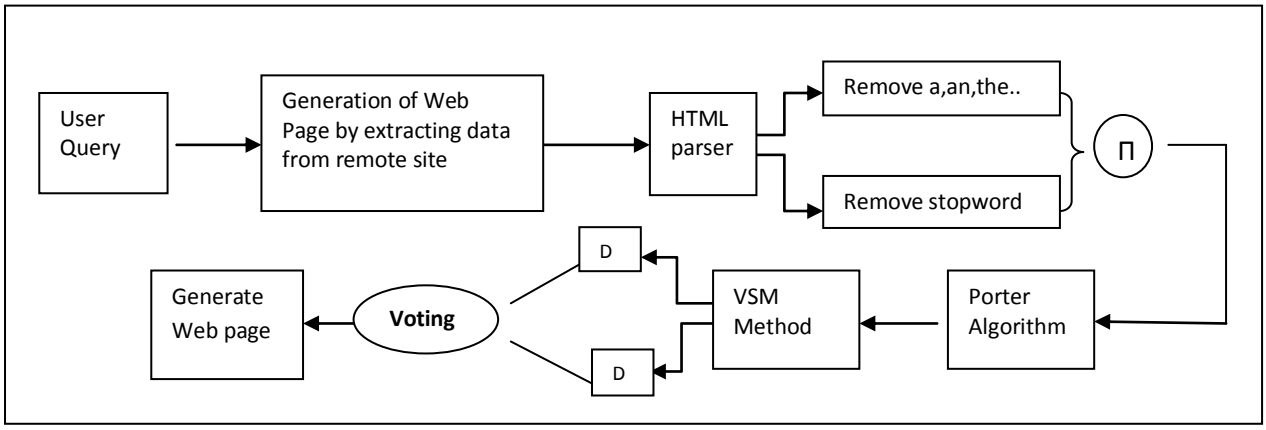

Fig1.Framework for content extraction and classification

4. WEB TEXT EXTRACTION

As the web pages on any website contains ample of unimportant data from user's point of view. Such as navigation bars, advertisements, various popup etc. But the content in which user is interested is little part of it. 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.[6].

4.1 Role of HTML parser

HTML Parser is a Java library used to parse HTML in either a linear or nested fashion. Primarily used for transformation or extraction, it features filters, visitors, custom tags and easy to use JavaBeans. It is a fast, robust and well tested package. The two fundamental use-cases that are handled by the parser are extraction and transformation.

4.2 Removal of stopwords

Typically these strings of characters have to be transformed into a suitable format. Usually the irrelevant words are filtered using a stop word list and reducing the word size by stemming. Stop word list are the words which does not have any relevant meaning for classification like a, an, and etc [1].

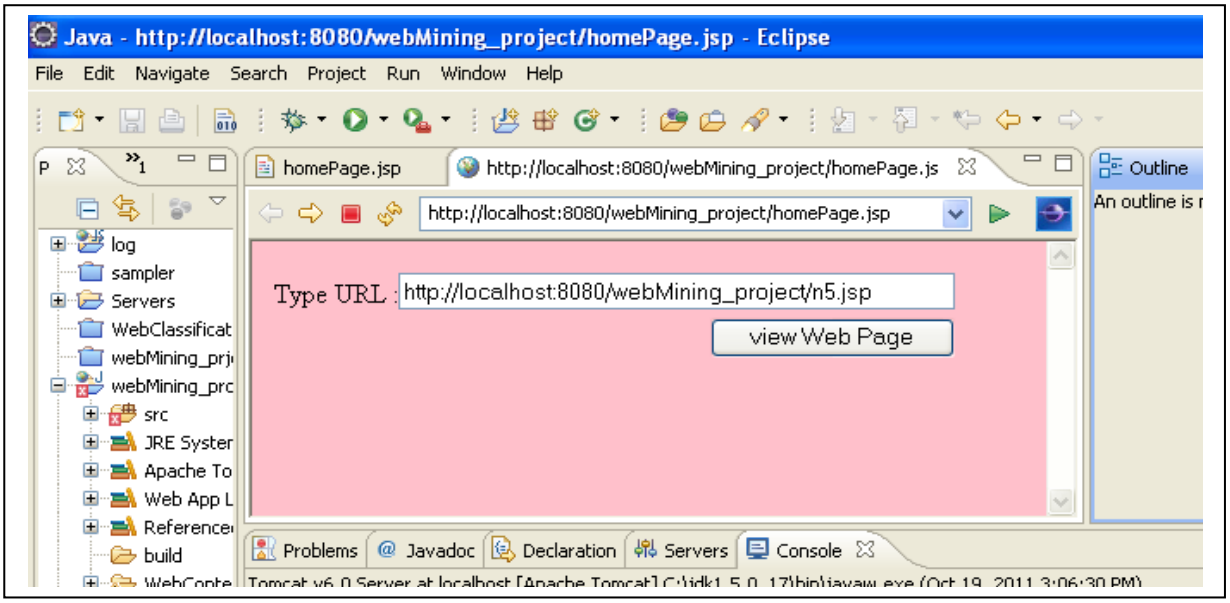

Fig. 2 Textbox for entering URL of website.

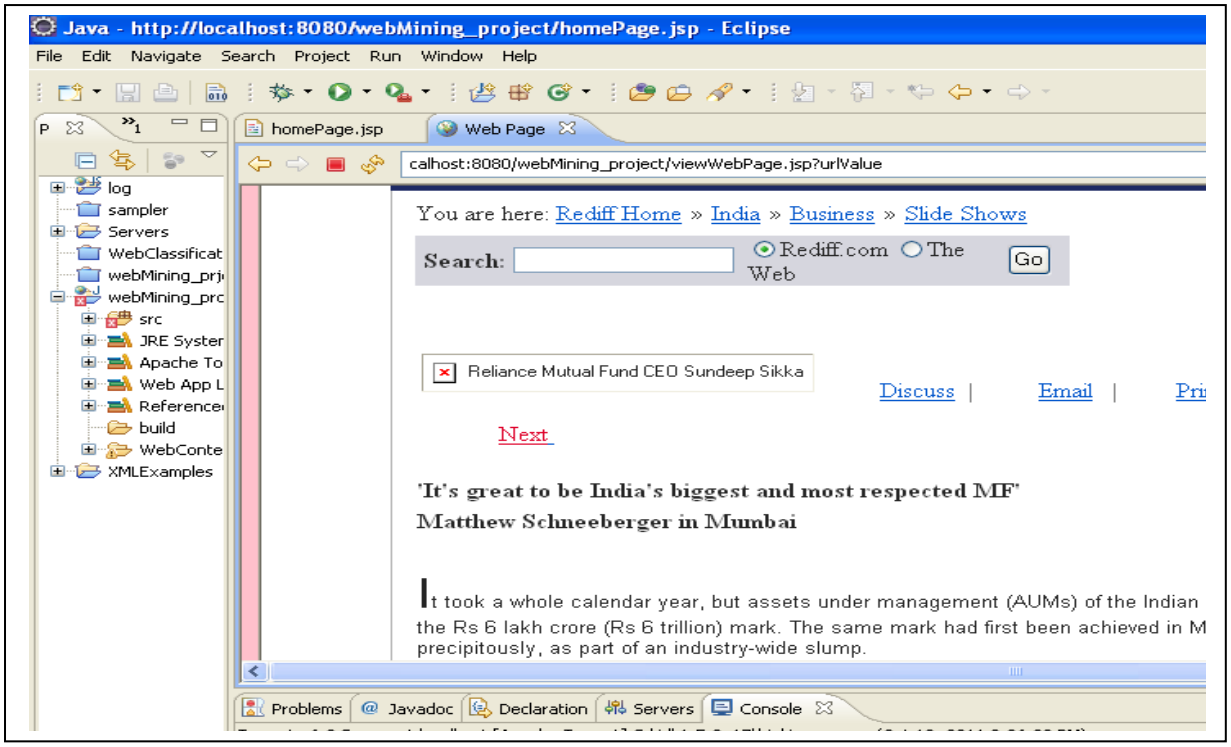

Fig. 3 Web page generation from URL of website.

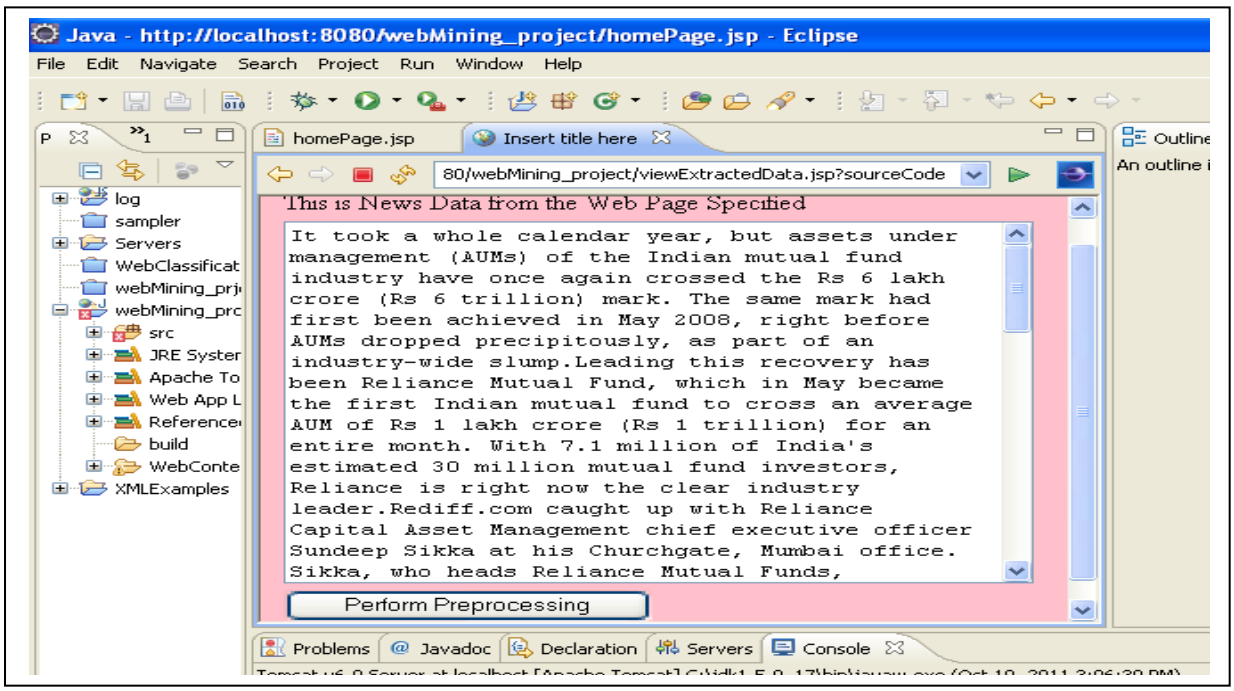

Fig. 4 Extracted data from webpage.

4.3 Stemming

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[7]. Fig.1 represent the user query which is given as address of website. Fig.2 gives the webpage for the address of website. It contains the images ,text etc. along with the main content. In the content extraction main contents will be extracted from the web page for the classification.

5. PORTER ALGORITHM

The algorithm is useful in removing suffixes and prefixes of the words. The algorithm follows given steps to output pure words which are useful for classification.

5.1 Step 1

1.Remove "es" from words that end in "sses" or "ies"– passes --> pass, cries --> cri.
2.Remove "s" from words whose next to last letter is not an "s"– runs --> run, fuss --> fuss.
3.If word has a vowel and ends with "eed" remove the "ed"– agreed -->agre, freed -->freed .
4.Remove "ed" and "ing" from words that have no other vowel– dreaded --> dread, red --> red, bothering --> bother, bring --> bring.
5.Add "e" is word has a vowel and ends with "ated" or "bled"– enabled --> enable, generated --> generate.
6.Replace trailing "y" with an "I" if word has a vowel–ex. satisfy -->satisfi, fly --> fly.

5.2 Step 2

With what is left, replace any suffix on the left with suffixon the right ex.-tionaltion conditional --> condition.

5.3 Step 3

With what is left, replace any suffix on the left with suffix on the right ex.-icateic fabricate --> fabric.

5.4 Step 4

Remove remaining standard suffixes al, ance, ence, er, ic,able, ible, ant, ement, ment, ent, sion, tion, ou.

5.5 Step 5

Remove trailing "e" if word does not end in a vowel ex-hinge-->hing.

6. VECTOR SPACE MODEL METHOD

After preprocessing, classification algorithm will be applied based on Vector space model (VSM) method in which similitude degree is used. The similitude degree method of literature search technique is adopted in the system for classification. Vector space model (or term vector model) is an algebraic model for representing text documents as vectors of identifiers. The vector space model procedure can be divided in three stages.[4]

1.The first stage is the document indexing where content bearing terms are extracted from the document text.
2.The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user.
3.The last stage ranks the document with respect to the query according to a similarity measure.

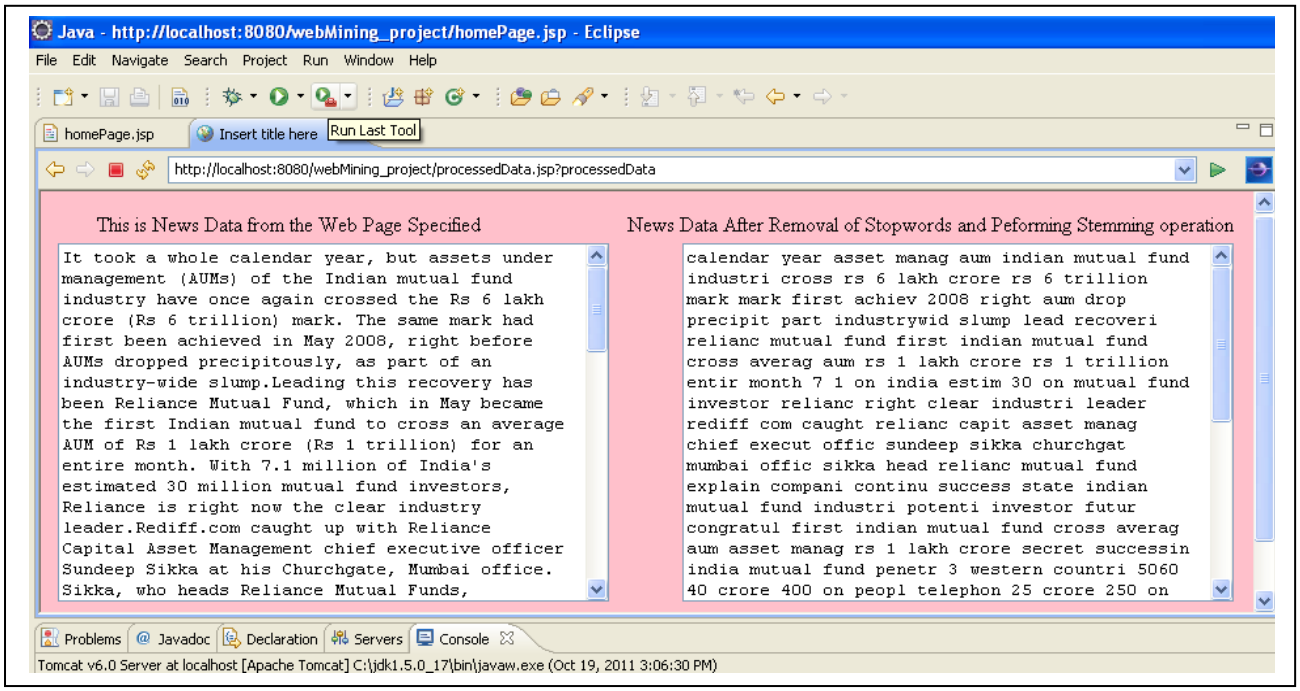

Fig. 5 Web page data after Preprocessing and using Porter Algorithm .

Salton's classic weighting is given by the following
Wi=tfi*log(D/dfi)
tfi = term frequency (term counts) or number of times a term i occurs in a document. dfi = document frequency or number of documents containing term i. D = number of documents in the database. For getting clear idea of similitude degree one example is explained. Suppose we query an IR system for the query "gold silver truck". The database collection consists of three documents (D = 3) with the following content.

D1:"Shipment of gold damaged in a fire"
D2:"Delivery of silver arrived in a silver truck"
D3: "Shipment of gold arrived in a truck"
We can write following information.
Q, D1, D2 and D3 are query, document1, document2 & document 3 respectively.
dfi specifies total number of occurrences of particular words in either D1 ,D2 & D3.
D/ dfi is total number of documents divided by dfi .
IDfi is the inverse log of D/dfi.

Weights of each Q, D1, D2& D3 is calculated by the product of counts and IDfi.

As $|Di| = \sum W^2 ij$

|D1|=0.7192 ,|D2|=1.0955,|D3|=0.3522

And as $|Q| = \sum W^2 Qj$

|Q|=0.5382

Next we compute all dot products.

Q.Di=∑(WQ j*Wi j)

Q.D1=0.1761*0.1761=0.0310

Q.D2=0.1761*0.1761+0.4771*0.9542=0.4862

Q.D3=0.1761*0.1761+0.1761*0.1761=0.0620

Now we will calculate the similarity values.

$$\text{CosineθD1} = \frac{Q.D1}{[|Q|*|D1|]} = \frac{0.0310}{[0.5382*0.7192]} = 0.0801$$

$$\text{CosineθD2} = \frac{Q.D2}{[|Q|*|D2|]} = \frac{0.4862}{[0.5382*1.0955]} = 0.8246$$

$$\text{CosineθD3} = \frac{Q.D3}{[|Q|*|D3|]} = \frac{0.0620}{[0.5382*0.3522]} = 0.3271$$

CosineθDi=sim(Q,Di)

$$\text{And sim(Q,Di)} = \frac{\sum_i W_{Q,j} W_{i,j}}{\sqrt{\sum_j W_{Q,j}^2} \sqrt{\sum_i W_{i,j}^2}}$$

Finally sorting and ranking of the documents is done in descending order according to the similarity values.

Rank 1: Doc 2 = 0.8246

Rank 2: Doc 3 = 0.3271

Rank 3: Doc 1 = 0.0801

Thus given query is closer to Doc 2 as a similarity value of Doc 2 is higher [8].

7. CLASSIFICATION

After computing vector length dot products are computed. In next step query is classified depending on similitude degree. Similitude degree between Documents(category) and Query decides the category of query. Threshold value is decided as 0.08.Condition for categorizing the query depending on threshold and similitude degree is as follows. This stage is categorization in which particular query is classified in particular category. For each category and training text vector length is computed [8]. If document 1 similitude degree is greater than document 2, document 3 than further is checked by threshold value. Similitude degree difference between threshold and similitude degree of document 1 is checked.

If similitude degree of document 2 is greater than that difference, query is multiclassified in 1st and 2nd document. Similarly it is checked for remaining documents. If none of the condition is satisfied then query belongs of only in 1st document. As shown above similar method is followed for document2, document3, etc. and threshold value. In last step user can do sub classification.

8. CONCLUSION

The framework described in this paper has produced good results for combinations of content extraction methods. The work is considered as extension to the existing traditional classification methods. Mathematical model for VSM method is explained with example. It has produced good results for classification with block diagram explained in paper.

9. REFERENCES

[1] Bing Liu ‘Web data mining’ Exploring hyperlinks contents and usage data.Springer Heidelberg, New York.

[2] Weiguo Fan1, Linda Wallace, Stephanie Rich, Zhongju Zhang “Tapping into the Power of Text Mining”.

[3] Suhit Gupta "context Based content Extraction of HTML Documents" M.S. Thesis Proposal, Dept of comp. sci.,Columbia University,New York,2004.

[4] Shiqun Yin Gang Wang Yuhui Qiu Weiqun Zhang. ” Research and Implement of Classification Algorithm on Web Text Mining”. IEEE.(2007)446-449

[5] Thomas Gottron. "Evaluatig content extraction on HTML documents" In ITA '07:Proceeding of 2nd International Conference on Internet Technologies and Applications, pages 123-132,September 2007.

[6] Neha Gupta, Dr.saba Hilal "A Heuristic Approach for Web content extraction"International Journal of Computer Applications(0975-8887) volume 15-No.5 Feb 2011

[7] Yin Yuhui Qiu Jike Ge, Xiaohong Lan.”Research and Realization of Extraction Algorithm on Web Text Mining”. (2007)278-281. Workshop on Intelligent Information Tech nology Application

[8] Shiquin Yin Yuhui Qiu ,Chengwen Zhong Jifu Zhou. “Study of Web Information extraction and Classification Method”.IEEE Transaction(2007)5548-5552.

[9] Yves Weissig, Thomas Gottron. “Combinations of Content Extraction Algorithms”.