Noise Reduction and Content Retrieval from Web Pages

Surabhi Lingwal
Computer Science & Engg.
G.B. Pant Engineering College
Pauri, Uttarakhand, India

ABSTRACT
The World Wide Web is the most increasingly growing and accessible source of information. Web contents of different fields which can offer important information to users are available in the Web like multimedia data, structured, semi-structured and unstructured data. But only a part of the information is useful for a particular application and the remaining information are considered as noises. Data on web pages contain formatting code, advertisement, navigation links, etc. This collection of unwanted noise with the real content in a web page complicates the task of automatic information extraction and processing. This requires the extraction of useful noise-free information. Otherwise, it can ruin the effectiveness of Web mining techniques. This paper proposes a novel method to filter web pages and retrieve the actual content of a web page. This research work proposed an approach for removing the noises from a given web page which will improve the performance of web content mining. At first, the web page information is divided into various blocks which then tokenized to separate the informative content from noises. This paper presents algorithm for removing noises from the web page and automatically extract important web content. This paper also presents the algorithm for global noise removal.

General Terms
Web Mining, Global Noises, Local Noises, Outlier, Document Object Model, Page Segmentation.

Keywords
Web content mining, content retrieval, noises, outlier, redundancy, precision, recall, accuracy**.**

1. INTRODUCTION
The rapid growth of the Internet has made the WWW a popular place for disseminating and collecting information. Users discover a lot of loaded hyper structure as the growth on the web is massive [4], [9], [13]. Updating incoming data and extracting relevant information without redundancy from the web quickly and efficiently becomes a growing concern among web mining research communities [6], [8]. Web content mining is the process of mining, extraction and integration of useful data, information and knowledge from Web page contents which has many applications, like it enable end users to access the web more easily over constrained devices like Personal Digital Assistants (PDAs) and cellular phones for to provide better access to the Web [18]. Web content mining includes a diverse kind of data such as: images, audio, video and texts. Core information, redundant information and hidden information are the three types of Web document data [14]. The content that a user needs to extract from a Web page is known as Core information like, in the news article Web page the main topic is the core information. In a Web mining system the input data moves through the three different stages to reach its final result: namely pre-processing, data mining and post-processing [3]. Pre-processing may include removing attributes that are irrelevant and cleaning the data from noisy information. Web noise is described as the information available in a Web page that is not relevant to the main content of the page. These noises are like banner commercials, navigational guides, garnishing images, etc. [1], [12], [19]. Before using the content of a Web page, the content is subdivided into smaller semantically homogeneous sections [16]. Web noises can be grouped into two categories according to their granularities [11].

1.1 Global noises
These are noises on the Web with large granularity; they are usually no smaller than individual pages. Global noises are like mirror sites, legal/illegal duplicated Web pages, old versioned Web pages to be deleted, etc.

1.2 Local (intra-page) noises

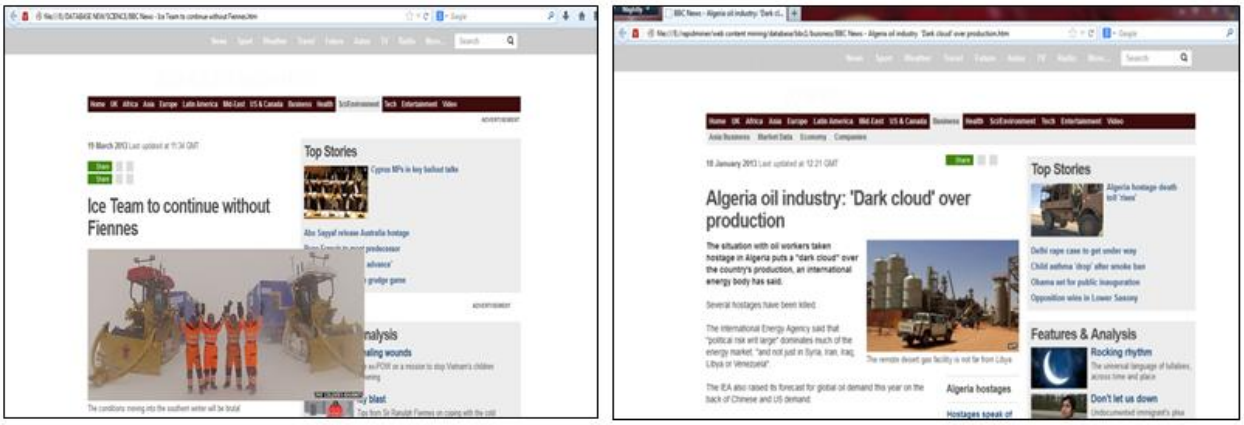

Fig 1: Two different web pages from BBC news site

These are noisy regions within a Web page. Local noises are usually incoherent with the main contents of the Web page. Such noises include banner advertisements, navigational guides, decoration pictures, etc.

This paper, first introduces a new technique to capture the actual contents of the pages in a Web site using RapidMiner [7]. This technique is based on term frequency and inverse document frequency of words in a document and based on it; noises are detected from the web pages. Secondly, it also detects the redundant data if it exists in a document database. Web content mining intends to mine valuable information or knowledge from Web page contents. The focus of web data extraction is to extract the contents from web pages for other applications like summarization, task learning, etc.

2. RELATED WORK

Recently, Akbar et al. [15] proposed an algorithm to extract the main blocks in blog posts. They adapted the Content Structure Tree (CST)-based approach by considering the attenuation quotient suffered by HTML block nodes. This is achieved by detecting primary and secondary markers for page clusters and then extracting the main block using these markers. However, Li et al. [2] have used VIPS to segment Web pages. They form a feature-vector considering visual, spatial, and content aspects of the extracted blocks to cluster them later. Gupta et al. [17] use a naïve approach to remove advertisements and links; their approach is based on the ratio of the number of linked and non-linked words in a DOM-tree. The linkage ratio in a block is a general indication that the block is likely to be part of the template, but this is not enough. For example, a block that has copyright information at the end of a page does not have high linkage ratio; however, it is part of the template. Fernandes et al. [5] proposed an approach specifically for information retrieval algorithms; it operates at block level. The authors claim that computing importance values of blocks based on the occurrence of terms in blocks (and not the whole page) gives much better results. It proposed a Noise Detector to identify blocks that make up a template in a website; this can later be used to infer the DOM structure of blocks that are part of the template. The latter step eliminates the overhead of segmenting Web pages and requires only traversing the Web page's DOM tree. Another advantage of Noise Detector is using semantically coherent blocks as the processing units to discover website's template. This makes Noise Detector unique in the way it detects templates where it uses blocks as the processing unit, and incorporates different aspects of Web pages such as: content, structure, and presentation. Derar Alassi et al. [6] proposed a Noise Detector (ND) as an effective approach for detecting and removing templates from Web pages. ND segments Web pages into semantically coherent blocks. Then it computes content and structure similarities between these blocks; a presentational noise measure is used as well. ND dynamically calculates a threshold for differentiating noisy blocks. Provided that the investigated website has a single visible template, ND can detect the template with high accuracy using two pages only. However, ND can be expanded to detect multiple templates per website, and the challenge will be to minimize the number of pages to be checked. Further, ND leads to website summarization. Jinbeom Kang et al. [10] proposed the RIPB(Recognizing Informative Page Blocks) algorithm that detects the informative blocks in a Web page by exploiting the visual block segmentation scheme. RIPB uses the visual page segmentation algorithm to analyze and partition a Web page into a set of logical blocks, and then groups related blocks with similar structures into a block cluster and recognizes the informative block clusters by applying some heuristic rules to the cluster information. The results of a series of experiments indicate that RIPB contributes to improve the accuracy of information extraction by allowing the wrapper induction module to focus only on the informative block information and ignore other noise information in building extraction rules.

3. PROPOSED WORK

The proposed work is carried out in RapidMiner. The content Extraction and Outliers Detection technique is based on term frequency and inverse document frequency of words. These words in a document are tokenized and relative frequency of occurrence in a document is calculated. This determines which words are considered as noises and which are considered as important word.

3.1 Outliers Detection

Outlier detection is a process in which the noises that are irrelevant to the main content are detected. In our algorithm first the documents are preprocessed for outlier detection. These documents are then tokenized to generate a list of words which is stored in a repository. It generates word vectors from a text collection stored in multiple files. The tokens of a document will be used to generate a vector numerically representing the document. Word vector is created based on TF-IDF value. The words are also pruned that are to frequent or to infrequent in a document. These words should be ignored for word list building. The word list from the repository is then retrieved, which is then transformed to data sets. This operation builds a data set from a word list. The data set contains a row for each word and attributes for the word itself, the number of documents in which it occurred, the number of labeled documents in which it occurred and for each class number it occurred in a document of this class. The operation is useful to filter a word list before reporting it. Outliers are identified in the dataset based on the distance to their k nearest neighbors. It utilizes a distance search through the k-th nearest neighborhood, so it implements some sort of locality as well. The method states, that those objects with the largest distance to their k-th nearest neighbours are likely to be outliers respective to the data set, because it can be assumed, that those objects have a sparser neighborhood than the average objects. As this effectively provides a simple ranking over all the objects in the data set according to the distance to their k-th nearest neighbours, therefore a number of n objects can be considered as the top-n outliers in the data set. This technique supports Euclidian distance which can be specified by a distance parameter. The algorithm takes a dataset and passes it on with an Boolean top-n D^k outlier status in a new Boolean-valued special outlier attribute indicating true (outlier) and false (no outliers).
Euclidian distance is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The Euclidean distance between points **p** and **q** is the length of the line segment connecting them ($\overline{\mathbf{pq}}$).

In Cartesian coordinates, if $\mathbf{p} = (p_1, p_2,..., p_n)$ and $\mathbf{q} = (q_1, q_2,..., q_n)$ are two points in Euclidean *n*-space, then the distance from **p** to **q**, or from **q** to **p** is given by

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \quad (1)$$

The position of a point in a Euclidean *n*-space is a Euclidean vector. So, **p** and **q** are Euclidean vectors, starting from the origin of the space, and their tips indicate two points.

3.1.1. Algorithm for Outlier Detection

Input: A web page from BBC site.
Output: Document retrieved without outliers.
begin

```
Preprocess(web_page)
{
 document ⇐web_page;
 extract_text (document);
 tokenize(document);
 create(wordlist);
 store(wordlist) in a repository;
}
 Retrieve(wordlist)
 {
 Convert(wordlist to data);
 Outlier_detection(n, k)
    {
        No_of_outlier = n; // no. of outliers to be
        detected initially from the wordlist.

        No_ of_ neighbors = k; //no. of neighbors
        with whom the  distance is calculated.
        p, q; // two points in a word list
        Euclidean_distance(p, q)
```

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

```
        //objects with the largest distance to their
        k-th nearest neighbours are outliers.
        outlier ⇐ larger_d;
        outlier = true;
        return(outlier_wordlist);
    }
 }
```

end

3.1.2. Filter Outliers

This operation takes a Dataset as input and returns a new DataSet including only the data that fulfill a condition. By specifying an implementation of *Condition* and a parameter string, arbitrary filters can be applied.

This parameter string is specified by the parameter parameter_string. Instead of using one of the predefined conditions, implementation can be defined with the fully qualified class name. For "attribute_value_condition" the parameter string must have the form attribute op value, where attribute is a name of an attribute, value is a value the attribute can take and op is one of the binary logical operators similar to the ones known from Java, e.g. greater than or equals. For "unknown_attributes" the parameter string must be empty. This filter removes all examples containing attributes that have missing or illegal values. For "unknown_label" the parameter string must also be empty. This filter removes all examples with an unknown label value.

3.1.3. Algorithm for Filtering Outliers

```
begin
    Filter_outliers(outlier_wordlist)
        {
          Condition_class = attribute_value_filter;
            Outlier = false; //all outlier having true
            value are removed and their value
            becomes false.
        }
end
```

3.2 Content Extraction

Content Extraction is a process of extracting the relevant information from a web page. Relevant information is the core information of a web page that a user needs to view. For example, the main content in the Web page of a news article is the core information. These documents are arranged in a directory and preprocessed for content extraction. Initially word vectors are generated from a text collection stored in multiple files. This results in Text extraction which ignores and discards the structural information like xml or html tags. The tokens of a document will be used to generate a vector numerically representing the document. The word vector is created using TF-IDF. The algorithm also provide the meta data information of the text like filename, date etc. The text are also pruned that occur outside a range given. The outliers or noises are removed from the web pages with the above algorithm and the core information is extracted.

3.2.1. Extract Content

This algorithm extracts content from a HTML document. This operation extracts textual content from a given HTML document and returns the extracted text blocks as documents.

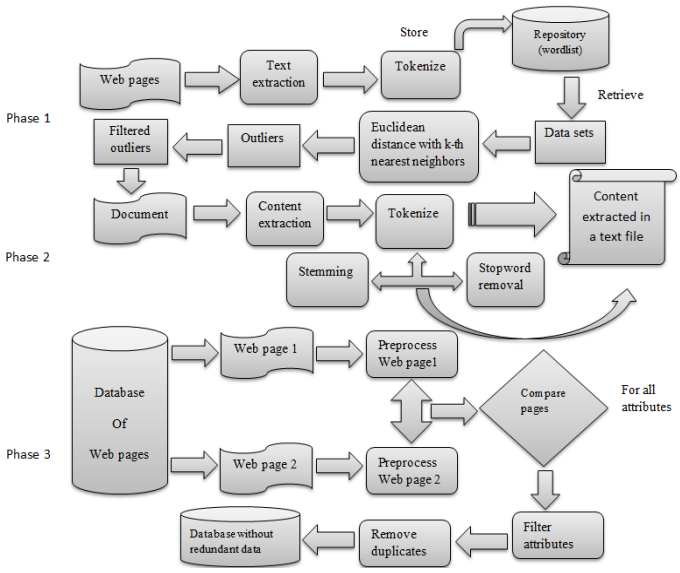

Fig 2: System Architecture of Proposed System

Only text blocks consisting of a given number of words are extracted to prevent single words (e.g. in navigation bars) to be kept. A minimum text block length is taken in terms of words or tokens. Some tags are chosen for content extraction like span tags, <p> tags, <b> tags, <i> tags,
 tags, and non html tags.

3.2.2 Tokenize

This operation splits the text of a document into a sequence of tokens. Splitting points may use all non-letter character; this will result in tokens consisting of one single word. Each non-letter character is used as separator. As a result, each word in the text is represented by a single token.

3.2.3 Filter Stopwords

This operation removes all tokens equal to a stopword from the given file. The file has to contain one stopword per line. Finally the text is written to a specified document file.

3.2.4 Algorithm for Content Extraction

Input: web page from BBC site
Output: Main content extracted in a text file
begin

```
Content_Extraction(web_page)
{
  Preprocess(web_page);
  Retrieve(wordlist);
  Outlier_detection(n, k);
  Filter_outliers(outlier_wordlist);
  Extract_content(document)
    {
      extract_text;
      minimum_text_block_length = 5;
       ignore_<b>;  ignore_<span>;
       ignore_<i>;  ignore_<br>;
      ignore_non_htmltags;
    }
  Tokenize(document)
        mode = non_letters;
  Stem_snowball(document)
        Language = English;
  Filter_stopwords(document);
  Write_document(file);
}
```

End

3.3 Redundancy Removal

This algorithm removes duplicate from the dataset by comparing all web pages with each other on the basis of specified attributes. Hence two examples are equal if all values of all selected attributes are equal.

3.3.1 Algorithm for Redundancy Removal

Input: database containing web pages
Output: database without redundant web pages
begin

```
Redundancy_removal(database)
{
  Preprocess(web_page1);
  Preprocess(web_page2);
  Compare(web_page1, web_page2)
    {
      if(attributes_web_page1 ==
      attributes_web_page2) then
        filter_attributes;
        Remove_duplicates;
    }
}
```

End

4. Results and Evaluation

This section evaluates the proposed outlier detection and content extraction algorithm. Since the purpose of our work is to extract the relevant information from a web page and store it in a database so it could be used in future for any information requirement. It also removes the redundant documents in a database. The accuracy of the algorithm evaluated to 85%.

4.1. Datasets

The experiment is conducted on 700-800 web pages taken from the various fields of BBC news site. The fields that are included are: science, technology, entertainment, health, and business.
BBC– http://www.bbc.co.uk/: news website

4.2. Results

4.2.1 Outliers Detection and Removal

The outliers are detected from the total words that occur in a document from the class BBC. This is shown in the figure 3 where the red dots indicate outliers whereas the blue dots indicate total words that occur in a web page. This graph plots the scatter color between words and the outliers in 3 dimensions. For outlier detection, outliers are assigned true value while words are assigned false value so that they can be detected accurately. Number of outliers to be found is given as 50 while number of nearest neighbors is initialized as 30.

The objects with the largest distance with the nearest neighbors are detected as outliers. The distance is measured through Euclidean distance.

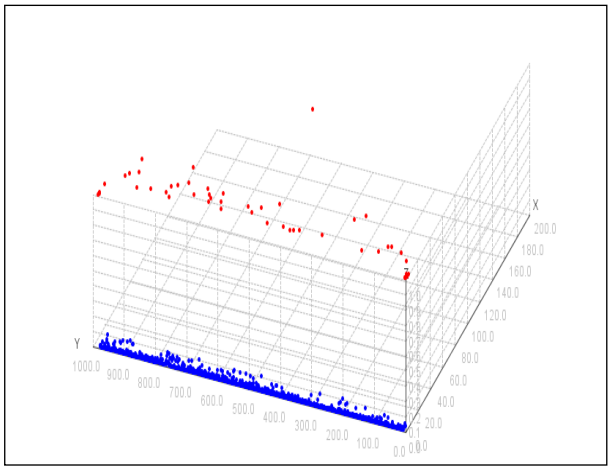

Fig 3: Outliers detected from the total words in a document

The detected outliers are then removed by filtering them. The graph shown in figure 4 represents data without noises or outliers. The blue dots represent total words in a document while the red dots shown in previous figure were removed. This filtered document is then used for content extraction for retrieving the main content from the web pages. This graph is shown in 3 dimensions where one dimension shows total, another one words and the next is for outliers.

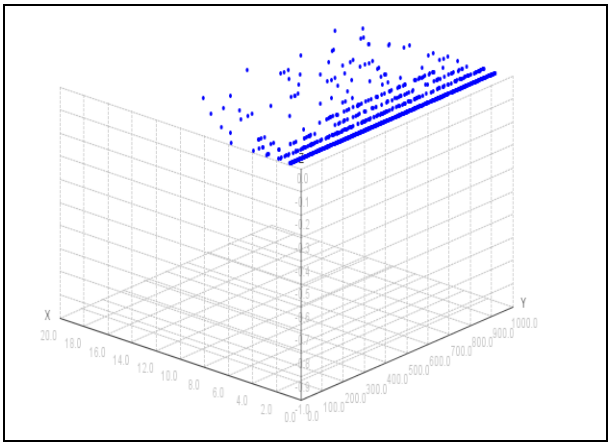

Fig 4: Outliers removed the from datasets

4.2.2 Content Extraction

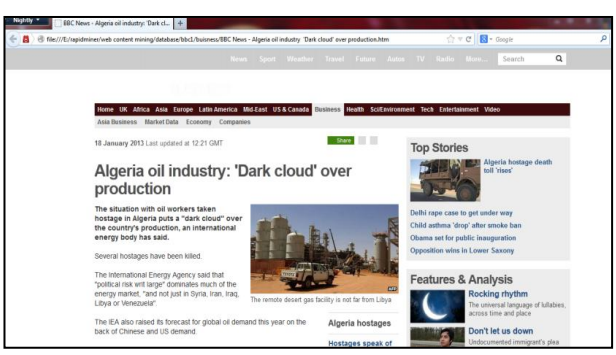

Fig 5 : A web page from business field of BBC new site

Core information extraction from a BBC site web page proceeds after outliers detection and removal. The figure 5 below represents a web page from business field of BBC news site.

The document retrieved after outlier removal is then processed for content extraction where content is extracted from html pages which are then tokenized, stemmed and stopwords are removed. Finally the text is extracted on a document file.

Content extracted from a web page in RapidMiner is shown in figure 6 below. The words in two different colors represent the tokenized words and these words are stemmed to their root word to reduce the disk space requirement. Stopwords are also removed like a, an, as, etc.

Fig 6 : content extraction from web pages

The content extracted from the web page is then retrieved in a text document, this is shown in figure 7 given below.

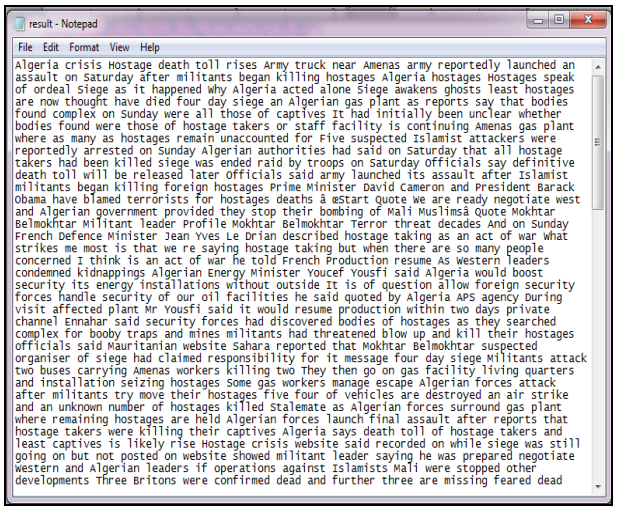

Fig 7: Content extracted from web page in text file

4.2.3 Redundancy Removal

Redundant data are the global noises that increase the burden over disk space. Therefore their occurrence should be reduced. The algorithm presented here, removes the possibility of near duplicates. If the database contains the same information or web pages in duplicity then it removes the duplicate web page or information and output the result after removing the duplicate data. This is shown in figure 8 below.

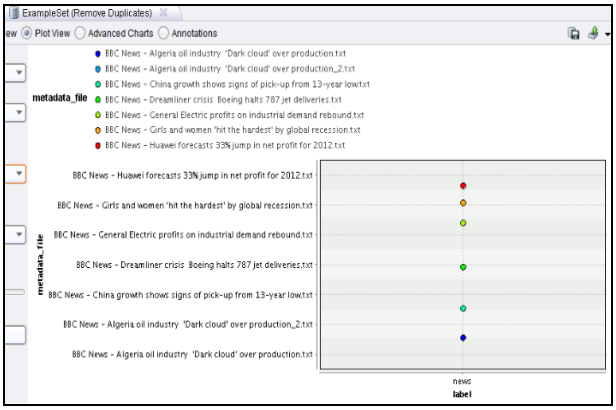

Fig 8 : Redundant data has been removed from dataset

4.2.4 Accuracy Measures

Precision: It is the ratio between the number of relevant documents returned originally and the total number of retrieved documents returned after eliminating irrelevant documents [8]. Here the relevant documents indicate the required documents which satisfy the user needs.

$$Precision = \frac{Relevant \cap Retrieved\ Originally}{Retreived\ after\ Refinement} \quad (2)$$

Recall: It is the ratio between the number of relevant documents returned originally and the total number of relevant documents returned after eliminating irrelevant documents [8].

$$Recall = \frac{Relevant \cap Retrieved\ Originally}{Relevant\ after\ Refinement} \quad (3)$$

The Content Extractor is capable of differentiating two classes of blocks, namely noisy and informative. A high sensitivity score (recall of the target class) means that the informative blocks have been well recognized; and a high specificity score (recall of the other class) means that the noisy blocks have been recognized. The calculated accuracy, precision and recall value is shown in table 1. and its graph is shown in figure 9.

Table 1. Accuracy measures for content extraction

BBC news area	Accuracy Measures			
	No. of Documents	Accuracy (%)	Precision (%)	Recall (%)
Science	150	80	85	80
Tech.	150	82	80	83
Health	150	85	85	80
Business	150	80	80	85
Enter.	150	85	82	85

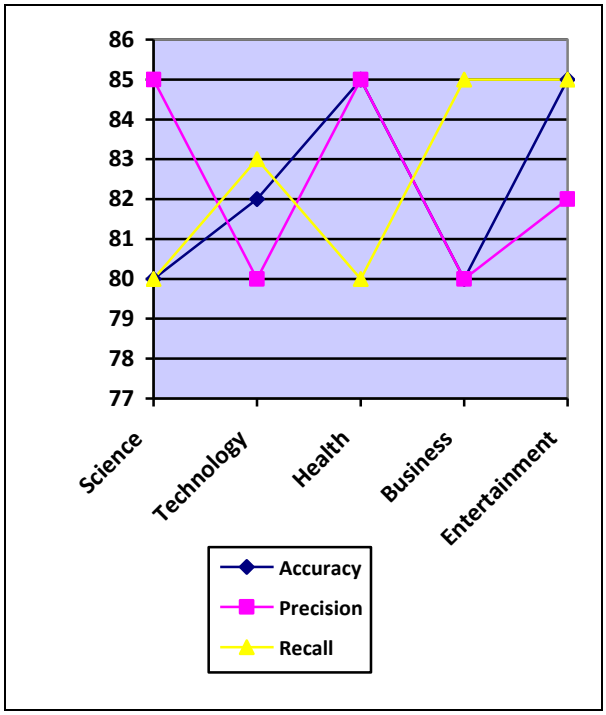

Fig 9: Graph for accuracy, precision and recall

5. CONCLUSION

The Content Extraction implemented in this paper could be seen as the result of realizing the importance of noise reduction in Web pages and converting the pages into more useful source of information especially for applications that require actual informative content of websites. The conducted experiments demonstrate that the approach described in this paper could be recognized as a major step in noise reduction. However, further improvements and extensions could lead to a more appropriate product.

The proposed Content Extractor applies robust noise measure to discover templates. The experiments conducted shows that the algorithms are able to detect outlier with high accuracy in websites. The experiments also show that removing noisy information, i.e., templates can improves the accuracy of Web mining tasks and content retrieval. Since the Web contains 40–50% noisy data, removing these noises becomes a necessity for better mining results. In fact, it not only works for local noises but also for global noises that reduces the storage space requirement. The experimentation reveals that about 85% of accuracy is achieved while extracting contents from web pages. Also the precision and Recall value is better than the previous approaches. The work can be further extended in other languages or in other software for better accuracy.

6. ACKNOWLEDGMENTS

A great thanks to all those people who guided and helped in this work for its successful completion.

7. REFERENCES

[1] A. K. Tripathy and A. K. Singh. 2004. An Efficient Method of Eliminating Noisy Information in Web Pages for Data Mining. In Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04), pp. 978 – 985, September 14-16, Wuhan, China.

[2] C. Li, J. Dong, J. Chen. 2010. Extraction of informative blocks from Web pages based on VIPS. Journal of Computational Information Systems 6 (1) ,271–277.

[3] D. Alassi, R. Alhajj. 2013. Effectiveness of template detection on noise reduction and websites summarization, pp 41-72, Information Sciences 219.

[4] D. Cai1, S. Yu, Ji-Rong Wen and Wei-Ying Ma. 2003. Extracting Content Structure for Web Pages based on Visual Representation. In Proceedings of the 5th Asia-Pacific Web Conference on Web Technologies and Applications, pp. 406-417, Xian, China.

[5] D. Fernandes, E. Moura, B. Ribiero-Neto, A. Silva, M. Goncalves. 2007. Computing block importance for searching on Web sites, in: CIKM 2007, pp. 165–174.

[6] D. Gibson, K. Punera, A. Tomkins. 2005. The volume and evolution of Web page template, in: International World Wide Web Conference, ACM, Chiba, Japan, pp.830–839.

[7] F. Akthar, C. Hahne,. 2012. RapidMiner 5 Operator Reference. August 2012. www.rapid-i.com.

[8] G. Poonkuzhali , G.V. Uma, K. Sarukesi. 2010. Detection and Removal of redundant web content through rectangular and signed approach, International Journal of Engineering Science and Technology ,pp 4026-4032, Vol. 2(9).

[9] G. Poonkuzhali, K. Thiagarajan, K. Sarukesi and G.V. Uma. 2009. Signed Approach for Mining Web Content Outliers. World Academy of Science, Engineering and Technology, Vol.56, pp. 820- 824.

[10] J. Kang and J. Choi. 2007. Detecting Informative Web Page Blocks for Efficient Information Extraction Using Visual Block Segmentation. International Symposium on Information Technology Convergence, IEEE.

[11] L. Yi, B. Liu, X. Li. 2003. Eliminating Noisy Information in Web Pages for Data Mining, SIGKDD .03, August 24-27, IEEE.

[12] L. Yi and B. Liu. 2003. Web Page Cleaning for Web Mining Through Feature Weighting. In Proceedings of the 18th International Joint Conference on Artificial Intelligence,Vol.18, pp. 43-50, August 09 - 15, Acapulco, Mexico.

[13] M. Agyemang, K. Barker and R. S. Alhajj. 2005. Mining Web Content Outliers using Structure Oriented Weighting Techniques and N-Grams. In Proceedings of the ACM Annual Symposium on Applied Computing, pp. 482-487, New Mexico.

[14] P. Sivakumar, R. M. S Parvathi. 2011. An Efficient Approach of Noise Removal from Web Page for Effectual Web Content Mining. European Journal of Scientific Research, pp. 340-351, Vol.50 No.3. http://www.eurojournals.com/ejsr.htm

[15] S. Akbar, L. Slaughter, Ø. Nytrø. 2010.Extracting main content-blocks from blog posts. In: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, pp. 438–443.

[16] S. Debnath, P. Mitra, N. Pal, and C. Lee Giles. 2005. Automatic Identification of Informative Sections of Web Pages, IEEE Transactions on knowledge and data engineering, vol. 17, no. 9.

[17] S.Gupta, G. Kaiser, D. Neistadt, P. Grimm. 2003. DOM-based Content Extraction of HTML Documents, ACM, Budapest, Hungary.

[18] S. Gupta, G.E. Kaiser. 2005. Automating Content Extraction of HTML Documents. World Wide Web: Internet and Web Information Systems, 8, 179–224, Springer.

[19] Z. Cheng-li and Y. Dong-yun. 2004. A Method of Eliminating Noises in Web Pages by Style Tree Model and Its Applications. Wuhan University Journal of Natural Sciences, Vol.9, No.5, pp. 611-616.