Enhancing Content Extraction from Multiple Web Pages by Noise Reduction

Ms. Shalaka B. Patil, Prof. Rushali A. Deshmukh

Abstract— With excessive growth of Internet, World Wide Web has become great source of Information which has been well known as big data repository consist of different variety of web pages with informative data as well as large amount of non-informative (noisy) data such as advertisements, navigational bars, copyright notices, etc. Although such noisy data is functionally useful, web users are unable to obtain more exact information is a critical issue of web-based information retrieval. Non-informative data not only degrades the web application performance but also reduces user's interest or can mislead user. As a result, retrieving the informative data from web pages which are cluttered with huge non-informative data has become difficult task. Hence reducing such noisy data from web pages has gain immense importance in early days. In this paper, we have developed Webpage Content Searching technique which focuses on retrieving user needed informative content by characterizing non-content as noise. Webpage Content Searching technique makes use of presentational and structural similarity along with content similarity which is calculated using FP-Growth algorithm to discriminate informative and non-informative data. Webpage Content Searching technique is effective for any number of web pages regardless of their domain.

Index Terms— Content Extraction, DOM Tree, FP-Growth Algorithm, Noise Reduction, Webpage, Web Mining, Vision-based Page Segmentation

1 Introduction

MOST of the web pages follow common structure to a greater or lesser extent. The web pages convey information through informative blocks which are comprised by non-informative blocks. The common examples of non-informative blocks are advertisements, copyrights, privacy notices, links, navigational panels, etc. These non-informative blocks may help web users to navigate as per their requirement and necessary for web functionality but they may also prevent user for concentrating on informative blocks effectively. Hence, such non-informative blocks are irrelevant to user required content and should be removed before web mining [7].

Noise is "irrelevant or non-informative data". Efficiently retrieving high-quality content from web pages is crucial part of web mining. Broadly, the webpage noises can be classified into two categories [3]:

1) *Global Noises:* These noises are no smaller than complete webpage. Global noises include mirror web sites, legal/illegal duplicated web pages, old version web pages, etc.
2) *Local (intra-page) Noises:* These are the noisy sections within the webpage. Local noises are usually inconsistent with the main content of the webpage. Such noises include advertisement, scroll bars, company logos, service channels, etc.

Webpage noise can be defined as the information provided by web site services which may be beneficial according to their views but may not useful for the web users who are looking only for the main content of webpage [8]. Web pages have been full of noise. Fig. 1 represents a sample webpage from PCMag web site. This webpage contains information regarding Samsung Galaxy Mega 2 (AT and T) cell phone. The main content (Segment 3 in Fig. 1) only occupies 1/3rd of the original webpage, and rest of the webpage contains advertisements, navigational links, magazine subscription, irrelevant articles, etc. Such kind of noisy information should be removed to enhance the performance of web mining.

We have developed a framework to retrieve user defined content from a large number of web pages. In this work, we are focusing on extracting and detecting the content required by web user through user query which will be the primary input to the system. The system will gather multiple web pages which strikes user specified input (keywords) from different domains. From these web pages, the user required information is retrieved using presentational and structural similarity along with content similarity (frequency) which is calculated using FP-Growth algorithm. The non-content data will be removed from each webpage and the content blocks with high frequency user required content are merged into single HTML page and will be viewed as output.

1.1 Motivation

In web mining, the knowledge discovery from web which contains 45% - 50% of noise is a very crucial task. Such amount of noise not only reduces the web application's efficiency but also diminishes user's interest. Hence, for getting only required contents from web pages we have developed Webpage Content Searching system.

The system provides a common layout for multiple pages of different domains i.e. domain independency to remove the noise and to generate user required information by manipulating presentational and structural similarity along with content similarity which is calculated using FP-Growth algorithm. As per the study [13], FP-Growth algorithm is an effective algorithm for generating frequent itemsets. FP-Growth uses complex data structure to reduce time and space complexity.

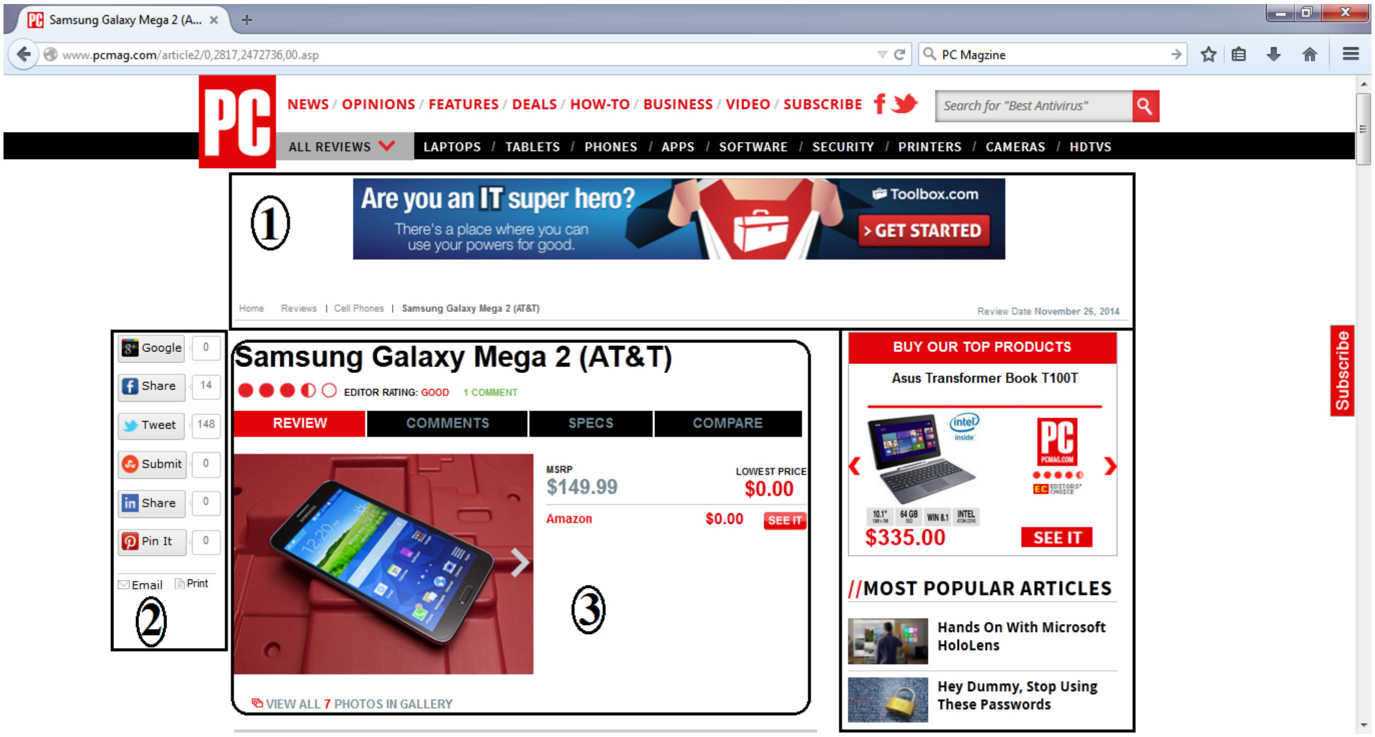

Samsung Galaxy Mega 2 (A...
www.pcmag.com/article2/0,2817,2472736,00.asp
PC Magzine
PC
NEWS / OPINIONS / FEATURES / DEALS / HOW-TO / BUSINESS / VIDEO / SUBSCRIBE
Search for "Best Antivirus"
ALL REVIEWS
LAPTOPS / TABLETS / PHONES / APPS / SOFTWARE / SECURITY / PRINTERS / CAMERAS / HDTVS
1
Are you an IT super hero?
There's a place where you can use your powers for good.
Toolbox.com
› GET STARTED
Home Reviews | Cell Phones | Samsung Galaxy Mega 2 (AT&T)
Review Date November 26, 2014
Google 0
Share 14
Tweet 148
Submit 0
Share 0
Pin It 0
Email
Print
2
Samsung Galaxy Mega 2 (AT&T)
EDITOR RATING: GOOD
1 COMMENT
REVIEW
COMMENTS
SPECS
COMPARE
MSRP
$149.99
LOWEST PRICE
$0.00
Amazon
$0.00
SEE IT
3
VIEW ALL 7 PHOTOS IN GALLERY
BUY OUR TOP PRODUCTS
Asus Transformer Book T100T
$335.00
SEE IT
//MOST POPULAR ARTICLES
Hands On With Microsoft HoloLens
Hey Dummy, Stop Using These Passwords
Subscribe

Fig. 1: Sample Webpage of PCMag Website with Noises (solid lines are drawn manually)

1.2 Issues and Challenges

Webpage Content Searching is not an independent research topic because the Webpage Noise Removal is dependent task which is always related to Web mining applications. Therefore, the categorization of Webpage noise removal and the Webpage content searching are two different critical tasks to improve the Web mining applicability and to help many Webpage content based tasks, e.g., information retrieval, information extraction, Web data warehousing, etc.

The main issue with existing Webpage cleaning methods is that they were not able to recognize or find all kind of Webpage noises to be removed. The most of the methods work only on some specific kind of noises (such as, advertisement, link list, images or hyperlinks etc.) to be removed for webpage cleaning, they worked only on general noises which are found on web pages. Some Web sites are structured with dynamic Web pages and their content and presentation style are not common. It is difficult to detect different noise structures for dynamic Web sites by using previous technique. So, the previous technique [8] is site dependent and performance is based on similarities of web pages.

Content Extraction arena represents new challenges related with above issues. We have designed a framework to overcome these issues that works effectively on most of the noises found on web pages to be removed and also works on web pages of different web sites that mean the technique is providing site dependency and performance is not based on similarities of web pages.

The main component of this paper is the Webpage Content Searching system as outlined above. Its different components are described in the following sections. The necessary background is also covered in order to have this document more self-contained. The remaining paper is organized as follows. Section II details some of the related work performed for the system. In addition, it provides an overview of the background. In Section III, we have described the Webpage Content Searching system implementation. Section IV shows and discusses the results obtained. Finally, Section V is conclusions and future work direction.

2 Related Work

In early days, web content mining and analysis has gain more importance in research society. The main goal of the paper is to detect user required content by marking other non-content as noise. Our work described in paper extends over with several web concerned research arena which consist of informative and non-informative block detection, noise detection and elimination, etc. Therefore, we will survey different papers which concentrate on detecting and removing noise from web pages using their own individual techniques/methods.

Shian-Hua Lin et al. [1] proposed a system named InfoDiscoverer which works only with HTML <table> tag. Using <table>, system first separates the whole webpage into different content blocks. This function performs for the HTML documents of the same site and forms Page Cluster [11]. After this, system calculates entropy value of every feature which occurs within the set of pages. Using this feature entropy value, the Content Block (CB) entropy is calculated. If this CB entropy value is higher than the defined threshold the content block is referred to non-content and if less, then the content block is informative.

Suhit Gupta et al. [2] addressed “Content Extraction” [9] solution for informative content mining which works with DOM trees instead of raw HTML markup. The solution works within two sets of filters. First set of filter simply filter out the tags which quickly eliminates images, links, scripts, styles and many more elements from webpage. But second set of filter is more algorithmic and complex. This set of filter consists of different remover applications such as the advertisement remover to remove advertisements; the link list remover which removes the list of links which are greater than defined threshold resides in table cells, the empty table remover that simply eliminates the table which does not consist of any substance, and the removed link retainer works distinct than all other removers. This retainer stores the deleted links for enabling users to access retained links.

Lan Yi et al. [3] deployed a new technique named as “Style Tree” [12]. The DOM trees were incapable for removing noise from web pages; hence, they proposed a technique Style Tree. The paper states that many of web pages follow similar structure for same website. The style tree merges all DOM trees of pages and calculates the node importance. The higher node importance is the non-informative and lesser node importance is informative block. But node importance can be calculated for single node also by using their featured attributes.

YuJuan Cao et al. [4] developed a method that utilizes visual features of the webpage. The visual features of the page are segmented using Vision-based Page Segmentation. According to the authors, typically web authors place the most important information in the middle of the page and provide navigational panels on the header, the left or right side of the page and copyright at the footer at last. Hence, they formed a feature vector to represent an informative block using 14 features of the page such as {Block_center_X, Block_center_Y, Block_width, Block_height, Font_Size, Font_Weight, Text_Proportion, Img_Num, Img_Size, Link_Num, Link_Text_Proportion, Anchor_Length, Para_Num, Negative_word}.

Yuancheng Li et al. [5] proposed a novel algorithm to construct a new tree i.e. Content Structure Tree (CST). The DOM tree is used with corresponding rules to construct Content Structure Tree. CST mainly consist of two types of nodes viz. namely HTMLItem Node and Content Node. HTMLItem nodes are generated using HTML tags such as BODY, DIV or TABLE, but Content nodes are formed with only text content. The Content Node Importance and HTMLItem node importance is used to identify the primary informative blocks from web pages.

Yan Guo et al. [6] provided a trivial but effective approach, named ECON (Extracting COntent from web News page). ECON uses backtracking to remove the noise from web news page. ECON first searches content snippet-node of news and then backtracks from snippet-node till a summary node is discovered. The key issues focused by ECON are: 1) How to discover one snippet-node to initiate backtracking? 2) When to terminate backtracking to find the summary-node? 3) How to eliminate noise during the backtracking?

Neetu Narwal [7] focused on the arena of noise removal from the webpage. The author has developed an algorithm which extracts Visual blocks of webpage in the form of DOM trees. These DOM trees are the converted into Pattern trees. For these pattern trees, the Node Importance and Style Importance are measured which classifies the informative and non-informative blocks. These measures are compared with predefined threshold and if measures are greater than threshold the assigned as main content otherwise assigned to noisy information.

Derar Alassi et al. [8] proposed a Noise Detector (ND) architecture which works in five main modules. Noise Detector first divides the whole webpage into blocks using Vision-based Page Segmentation. This module divides DOM tree of page into small DOM trees for each block. The blocks are then filtered using content threshold 20. If content are less than 20 words, the block is eliminated. After the filtration the noise matrix is formed. For noise matrix formation, content and structure similarity is calculated along with the presentational noise measure. Noise Detector dynamically computes a threshold for differentiating noisy blocks. Author’s related work demonstrates that an examined website has a single visual template; Noise Detector is able to detect highly accurate template using two web pages only. However, Noise Detector can be extended to find multi-templates per website, and the challenge will be reducing the number of pages to be checked.

We cover from the literature, the main approach which detect the noise and simultaneously removes them from web pages.

3 Implementation Details

3.1 Problem Definition

Web pages are designed to transfer the information through informative blocks which are highlighted with the main content. Rather than this main content block, web pages are surrounded of huge non-informative blocks such as pop-up ads, service logos, unnecessary images, navigational panels and lots of extraneous links [5]. These non-informative blocks not only degrade the performance of web mining applications as well they can misguide the web user which is more misfortunate. Hence, these non-informative blocks must be removed for improving web mining.

3.2 Webpage Content Searching System Architecture

An exact user needed informative block [10] as an output determines the quality and effectiveness of the system. Webpage Content Searching technique, according to problem definition consists of following steps:

1) *Webpage Extraction:* Various web pages of different websites have their distinct template which means different

layouts for different web pages. This arise the situation of multi-templates. The proper extraction of webpage from the domain site is an important factor i.e. webpage's content, structure and presentational data must be fetched completely. On a client's request, web pages are fetched from server side and web server will respond with HTML files for valid web pages.

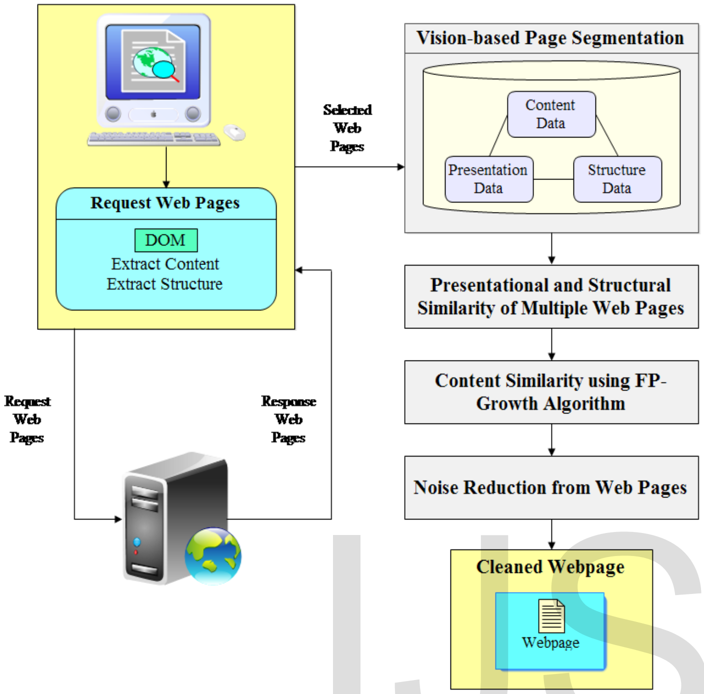


Fig. 2: Searching Informative Blocks from Web Pages

2) *Vision-based Page Segmentation:* After extracting the HTML files of web pages which are represented to the system in the form of DOM tree of whole HTML page, they are segmented using Vision-based Page Segmentation [4] technique into different dataset viz. namely content dataset, structure dataset and presentational dataset which contains actual text content of webpage, the body tags required to format the content and the head tags which resides for presentational view of whole webpage respectively.

3) *Similarity between Multiple Web Pages:* After forming content, presentation and structure datasets, we'll manipulate those datasets to calculate similarity between selected multiple web pages one after one. Presentational similarity will represent the web pages of same website. Structural similarity will count the similar tags of all selected web pages. And finally, content similarity is calculated using FP-Growth algorithm for every <DIV> tag of web pages. As per study, DIV tags consist of main informative content of webpage. If DIV tag's frequency is greater than predefined threshold then those tags are discriminated as informative blocks of a webpage, combined into single webpage, and displayed as output result to the user.

Hence, the success definition for the system is that the system is capable of merging multiple webpage's high frequency informative blocks of different web sites in a single refined webpage.

The architecture of Webpage Content Searching system depicts multi-threaded architecture. When multiple web pages are fetched into a common layout, each webpage is treated as a thread and the similarity measures are calculated for each selected webpage simultaneously which makes the architecture multi-threaded. Fig. 3 shows the flow of Webpage Content Searching system.

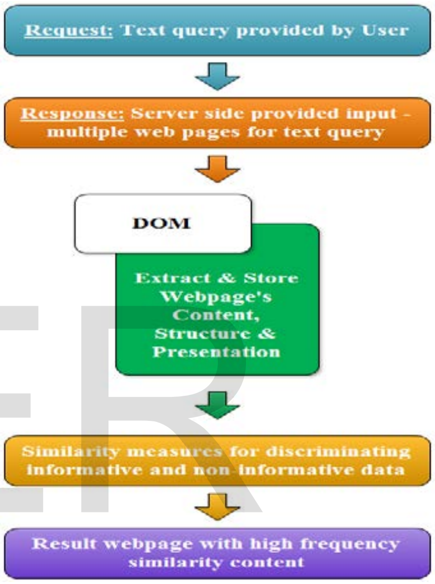

Request: Text query provided by User
Response: Server side provided input - multiple web pages for text query
DOM
Extract & Store Webpage's Content, Structure & Presentation
Similarity measures for discriminating informative and non informative data
Result webpage with high frequency similarity content

Fig. 3: Data Flow Architecture

3.3 Algorithm

Webpage Content Searching (WCS) algorithm will reflect the detail work of the system design after providing a text query as a primary input. Algorithm shows the classification of HTML file into three separate segments viz. namely presentation, structure and content based on their data type. The content within DIV tags are compared with the text query of the user. If the content is similar to the text query within the webpage, it will be stored in Similarity Dataset. Using this dataset, system will search for frequent occurrence of text query in each DIV for extracted webpage and calculates similarity (frequency) within those DIV tags of each page for user required content. The DIV tags having frequency greater than or equal to already defined threshold will be reconstructed within common panel and presented as new created webpage to user as an output.

```
Input: $txt$ = user.searchquery
$Initial$ WebpageSet $P = \{P_1, P_2, P_3 \ldots P_n\}$
pageData = extractWebpageSource([$P_i$])

Output: newCreatedWebpage
```

Algorithm: WebpageContentSearching(***txt,P***)

```
contentofpage = null;
structuretagset = null;
presentationtagset = null;
SimilarityDataset = null;
newstructureset; //Stylesheet for illuminating output

for pageData.length != null
{
        if(find head_tags in pageData)
        Presentationtagset = pageData[head_tags];
        //Presentation_Dataset for storing tags of web pages
        else if(find body_tags in pageData)
        structuretagset = pageData[body_tags];
        //Structure_Dataset for storing tags of web pages
        else
        contentofpage = pageData[content];
        //Content_Dataset for storing content of web pages
}

for each DIV in pageData
 if(DIV[content] == txt)
{
add DIV[content] in SimilarityDataset;
Using FP-Growth algorithm find frequency for
  DIV[content];
}

for get index from contentofpage
{
  if(f(DIV[content])≥threshold)
        //Occurrence(frequency) of user required content ≥ threshold
  {
  set contentstructuretag[index] = add contentofpage
  into newstructureset; //Output with informative blocks
  ContentSet = contentstructuretag[index];
  }
  else
  {
  set noisestructuretag [index] = add contentofpage
  with newstructureset; //Noise blocks per webpage
  Non-ContentSet = noisestructuretag[index];
  }
}
newCreatedWebpage ← ContentSet
return newCreatedWebpage
```

3.4 Experimental Setup

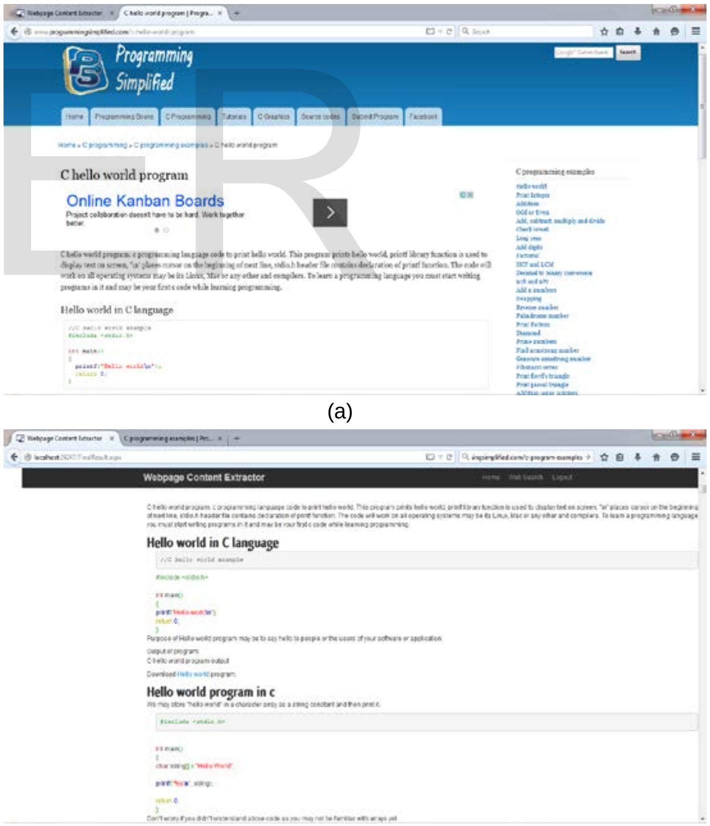
The system has been developed using C#.Net programming language. A graphical user interface has been designed using ASP.Net to make it more interactive and user friendly. The experiments have been conducted on a machine with the following specifications: Intel i5 Quad Core Processor with 2410MHz CPU, 4GB RAM, and running Microsoft Windows 7 Ultimate 64-bit operating system.

4 Experimental Results

We present here the experimental results to testify the effect of algorithm. In the evaluation process, we have used different web pages of different websites. The web pages are selected by user for search query "programming in c". The following five web pages have been used in the evaluation process:

1) http://www.programiz.com/c-programming
2) http://www.programiz.com/c-programming/c-keywords-identifier
3) http://www.programiz.com/c-programming/examples
4) http://www.programiz.com/c-programming/c-functions
5) http://www.programmingsimplified.com/c-program-examples

Fig. 4 is a sample webpage about "programming in c " which come from tutorial website (Last Webpage from above list). In this example, there are so many links are present in the web-page which is part of main content and they can produce enough noise. The block of recommended reading is in the main content, but the block is contained in the div tag, besides, it also has the link text, this noise block is trimmed in the pre-processing.



C hello world program

C hello world program: c programming language code to print hello world. This program prints hello world, printf library function is used to display text on screen, '\n' places cursor on the beginning of next line, stdio.h header file contains declaration of printf function. The code will work on all operating systems may be its Linux, Mac or any other and compilers. To learn a programming language you must start writing programs in it and may be your first c code while learning programming.

Hello world in C language

```
//C hello world example
#include <stdio.h>

int main()
{
  printf("Hello world\n");
  return 0;
}
```

(a)

Webpage Content Extractor

C hello world program: c programming language code to print hello world. This program prints hello world, printf library function is used to display text on screen, '\n' places cursor on the beginning of next line, stdio.h header file contains declaration of printf function. The code will work on all operating systems may be its Linux, Mac or any other and compilers. To learn a programming language you must start writing programs in it and may be your first c code while learning programming.

Hello world in C language

```
//C hello world example
#include <stdio.h>

int main()
{
printf("Hello world\n");
return 0;
}
```

Purpose of Hello world program may be to say hello to people or the users of your software or application.

Output of program:
C hello world program output

Download Hello world program.

Hello world program in c

We may store "hello world" in a character array as a string constant and then print it.

```
#include <stdio.h>

int main()
{
char string[] = "Hello World";

printf("%s\n", string);

return 0;
}
```

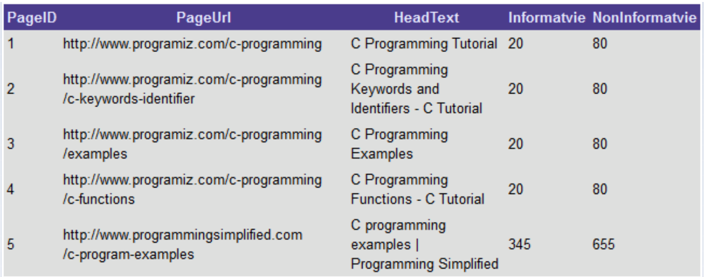
Don't worry if you didn't understand above code as you may not be familiar with arrays yet.

b)

Fig. 4: (a) A Sample Webpage (b) The Result after Content Extraction

Table I shows the web pages along with their paths. The remaining columns display Informative & NonInformative Number of Blocks of WebPage. The results are generated by considering the major parent tags of respective web pages.

TABLE I: Page URL along with title of Webpage and Informative & NonInformative Number of Blocks of Webpage



PageID	PageUrl	HeadText	Informatvie	NonInformatvie
1	http://www.programiz.com/c-programming	C Programming Tutorial	20	80
2	http://www.programiz.com/c-programming /c-keywords-identifier	C Programming Keywords and Identifiers - C Tutorial	20	80
3	http://www.programiz.com/c-programming /examples	C Programming Examples	20	80
4	http://www.programiz.com/c-programming /c-functions	C Programming Functions - C Tutorial	20	80
5	http://www.programmingsimplified.com /c-program-examples	C programming examples | Programming Simplified	345	655

The given chart shows the graphical results for content and non-content among web pages. Fig. 5 represents web page along with Informative and Non-Informative number of blocks. It depicts webpage ID on horizontal axis and the number of blocks depicted by vertical axis.

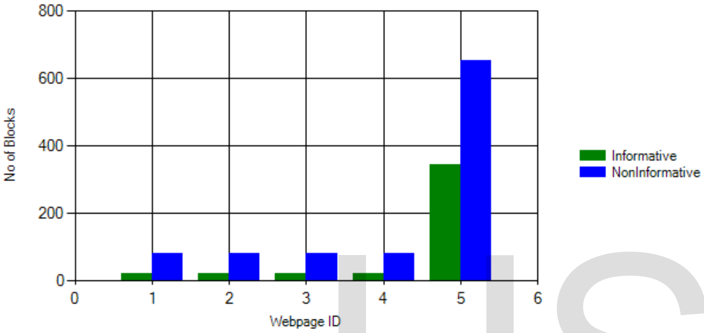


Fig. 5: Webpage Number vs. Number of Informative and Non-Informative Blocks

5 Conclusion and Future Work

In this paper, we employed with Webpage Content Searching technique which focuses majorly on retrieving user required content by assigning all other non-content as noise from HTML documents of web pages. The HTML file of webpage is divided into distinct datasets using Vision-based Page Segmentation. Webpage Content Searching technique uses content similarity measures which are calculated using FP-Growth algorithm to discriminate informative and non-informative data. The benefit of Webpage Content Searching technique over the existing system is that any number of web pages which do not belong to same domain can be used for content extraction. Thus, finally we conclude that Webpage Content Searching technique extracts similar contents from different web pages of different websites without noise.

As we are combining multiple web pages' informative blocks in a single HTML page, the same (redundant) informative blocks can appear on page which belongs to different domain web pages. This redundancy can be removed by analyzing final output webpage which is the future direction for the paper. In this paper, we are extracting images only using their outer HTML text. Hence, the images appear in web pages can also consist of information which must be read using image processing techniques to assign them as informative or non-informative.

References

[1] Shian-Hua Lin, Jan-Ming Ho, "Discovering Informative Content Blocks from Web Documents", Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 23-26, 2002, Page(s): 588-593.

[2] S. Gupta, G. Kaiser, D. Neistadt, P. Grimm, "DOM-based Content Extraction of HTML Documents", Proceeding WWW '03 Proceedings of the 12th international conference on World Wide Web, 2003, Page(s): 207-214.

[3] Lan Yi, Bing Liu, Xiaoli Li, "Eliminating Noisy Information in Web Pages for Data Mining", Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, August 24-27, 2003, Page(s): 296-305.

[4] YuJuan Cao, ZhenDong Niu, LiuLing Dai,YuMing Zhao, "Extraction of Informative Blocks from Web Pages", Advanced Language Processing and Web Information Technology, 2008. ALPIT '08. International Conference on 23-25 July 2008, Page(s): 544 – 549.

[5] Yuancheng Li, Jie Yang, "A Novel Method to Extract Informative Blocks from Web Pages", Artificial Intelligence, 2009. JCAI '09. International Joint Conference on 25-26 April 2009, Page(s): 536 – 539.

[6] Yan Guo, Huifeng Tang, Linhai Song, Yu Wang, Guodong Ding, "ECON: An Approach to Extract Content from Web News Page", Web Conference (APWEB), 2010 12th International Asia-Pacific, 6-8 April 2010, Page(s): 314 – 320.

[7] Neetu Narwal, "Improving Web Data Extraction by Noise Removal", Communication and Computing (ARTCom 2013), Fifth International Conference on Advances in Recent Technologies, 20-21 Sept. 2013, Page(s): 388 – 395.

[8] Derar Alassi, Reda Alhajj, "Effectiveness of template detection on noise reduction and websites summarization", Information Sciences 219 (2013) 41–72.

[9] Alpa K. Oza, Shailendra Mishra, "Elimination of Noisy Information from Web Pages", International Journal of Recent Technology and Engineering, March 30, 2013, Page(s): 115 – 117.

[10] Yi-Feng Tseng, Hung-Yu Kao, "The Mining and Extraction of Primary Informative Blocks and Data Objects from Systematic Web Pages", Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on Dec. 2006. Page(s): 370 – 373.

[11] Xuan WANG, WeiPing WANG, Bowen LIU, Zhen WANG, Xicai WANG, "A Novel Approach To Automatically Extracting Main Content of Web News", E-Business and Information System Security, 2009. EBISS '09. International Conference on 23-24 May 2009, Page(s): 1 – 4.

[12] R. Gunasundari, S. Karthikeyan, "Removing Non-informative Blocks from the Web Pages", Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on 7-9 Oct. 2010, Page(s): 810 – 814.

[13] Gagandeep Kaur, Shruti Aggarwal, "Performance Analysis of Association Rule Mining Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering 3(8), August - 2013, Page(s): 856-858.