The amount of information available on web is increasing day by day and has reached its ultimate form in any time of history. This store of information is of great use for researchers and general public. The maximal use of this information can be done only if applicable tools are developed to extract and handle this large information. The problem is that the development of tools and techniques to extract and process this massive and diverse information has greater challenges as the web pages contains clutter (such as ads, unnecessary images and extraneous links) around the body of an article, which distracts a user from actual content according to the method described by (C.Mantratzis, et al.,2004). This mix of unwanted noise and clutter with the real content in a web page complicates the task of automatic information (content) extraction and processing. The term “content extraction” also called as “Information Filtering” means extraction of useful noise-free information from the web pages.

The content extraction is progressively applied in various fields and applications. Content Extraction is beneficial for visually impaired and blind by identifying the real content within a web page and then increase the font size of the portions of the web page containing contents for better visualization or directly transforming the contents of the web page to speech. The content extraction is used in fields of Natural Language Processing (NLP) and information Retrieval (IR). Where these models derive accurate results based on relevance of contents and the reduction of “standard word error rate (S.Gupta, et al.,2003). Most of the NLP based IR applications necessitate dedicated extractors for each of the web domain (S.Gupta, et al.,2003) . The generalized content extractors are sufficient and less laborious than hand tailored extractors but is often found less accurate (S.Gupta, et al.,2003). The field of Open Source Intelligence extracts the information from the web and automatically processes it to gain knowledge uses content extraction. The expansion of the information on the web and increasing number of web pages introduces the need for new tools and techniques for content extraction in accurate way. A generalized traditional web page contains a title banner, list of links in right or left

INTRODUCTION

The information available today on web is tremendous and comes with greater challenges. Content extraction identifies the main content and removes the clutter from web pages. It combines approaches and techniques like statistical features extraction, formatting characteristic. Content type identification is used along with collective approach to overcome problem of dealing with versatile web pages, and yielding to achieve more accuracy in extracting the contents. Combinational scheme for efficient content extraction from web pages uses a hybrid model that operates on Document Object Model (DOM) tree of the corresponding HTML document to extract the content accurately. The proposed hybrid model is derived from two different models based on Statistical features and Formatting characteristics. The model works on Document Object Model (DOM) tree of the corresponding HTML document. It evaluates each tree node and calculates the associated statistical features like Deviation, Normalized Deviation, Link Density and Normalized Link density to predict significance of the node towards overall content provided by the document. We have proposed a Combinational scheme for efficient Content extraction from web pages which determine the context of the web page and calculate its statistical feature, determine the useful nodes and extract the contents. The proposed approach increases the overall performance (i.e) accuracy of content extracted from web pages. The proposed system overcomes the disability to detect the web page type or context of web page and the difficulty in discarding relevant linkages from web pages.
or both for site navigation and advertisements, a footer containing copyright statements, disclaimers or even sometimes navigational links (T.Weninger, et al., 2010). The recent web pages tend to have more cleaner architecture using various layers for visual presentation, real content and interaction (T.V.Raman, 2009) having abandoned the use of old structural tags and adopted an architecture that makes use of the style sheets and div or span tags (T.Weninger, et al., 2010). This change in architecture eases the development process but complicates the extraction process thereby reducing the effectiveness of old content extraction systems. The old content extraction systems operate on any varied web pages without considering the type of content the web page represents, which in return yield less accuracy in the content extraction models. This paper proposes methodology to improve accuracy by identifying the content type.

**Efficient Content extraction Using Hybrid Technique:**

The paper proposes a hybrid model named as “Info Optics” (IO). This approach is called hybrid because it mainly operates on two models of content extraction one based on statistical features and other on Formatting characteristics. It functions on DOM tree representation of web page calculating the different statistical features associated with the different nodes of tree to measure their importance in providing the information. The different statistical values like text density and link density for each node are calculated. These values are normalized, so important content should be retained. The calculation is based on the fact that the nodes associated with the content have higher values for the quantity of the text and lower values for quantity of hypertext.

In order to achieve an optimal performance on different styled WebPages quantities are normalized with respect to each page. Once the statistically useful nodes are identified, other nodes similar to useful nodes based on formatting characteristics and their position in the page are identified. All of the nodes classified as useful and nodes similar to useful nodes are considered to be the nodes containing real contents. The proposed model identifies the layout of the page by: comparing the quantity of text across each unique node \(ø(i)\) in the DOM tree with the arithmetic mean of the quantity \(\text{Avg } ø(T)\) of text across all of the nodes in DOM tree. The deviation \(D(i)\) in text quantity at each node from the arithmetic mean signifies how much contribution of node to the information being rendered to the user. The higher the deviation, the more information is rendered through that node.

Use the following Eq(3.1) for calculation of text deviation \(D(i)\),

\[
D(i) = ø(i) - \text{Avg } ø(T)
\]  

where \(T\) is a set of all nodes in DOM tree of the document and \(i \in T\). \(ø(i)\) denotes the quantity of text at a particular position \(i\) in \(T\). Eq(3.1) will yield lower values for the nodes which have smaller quantities of text associated with them. To normalize the result of Eq(1) to a closed interval \([0, 1]\), use Eq(2).

\[
N(i) = \frac{D(i) - \text{Min}(D(T))}{\text{Max}(D(T)) - \text{Min}(D(T))}
\]

The normalized deviation \(N\) obtained from eq.(2) identifies the less informative nodes of DOM tree. The less informative nodes will yield very low values while the nodes which are used in rendering the most of the information have values nearer to 1.

However this is not sufficient to distinguish them from the informative nodes of DOM tree that have less quantity of information as for example the titles of the columns in tables, headings. To make this distinction, calculate the link densities related with the nodes of DOM tree along with their deviations.

\[
L(i) = \frac{L(i)}{ø(i)}
\]

where \(L(i)\) is the quantity of anchor text. The link density at a particular node can be divided by maximum link density within the page to normalize the results this can be calculated using Eq.(4).

\[
NL(i) = \frac{L(i)}{\text{Max}(L(T))}
\]
By comparing the $NL(i)$ with $N(i)$ at any DOM node, it can be identified that whether the DOM node contributes towards information or it is there for navigation. For the advertisements, headers or footers in any page, the link densities are usually high values. This behaviour is mainly because of the fact that the quantity of text within the DOM nodes which are part of headers, footers or advertisements is less and the quantity of links is more than the informative nodes of the DOM tree of the same page. In this module the web page is given as input and the contents of the meta tag are extracted to classify the website for the genre. If the website description matches the list of pre-classified genre of websites then the content extraction is done accordingly for that website. Example, if the website is of news domain then the content extraction has to be done only for the article in the website rather if the website is a shopping site too much removal of the links would render no useful information to the user.

**Architecture for Content Extraction System:**

All the website available on internet have information from different domain and need different strategy to achieve accuracy in extracting the contents. The previous approach used only fixed threshold for all the web pages which lead to extraction of more contents or sometimes less content. This approach sometimes extracted the useful information as for each web page the ratio of extraction should be set differently. Thus context extraction is used to achieve more accuracy in working with various web pages. The context extraction process involves extracting the context of the web page to perform the content extraction more accurately.

Fig.1 shows the architecture of context extraction along with the old system. In this the web page will be given as input to the module of Content genre determination. This module will have the list of top 200 pre-classified websites, the new web page will be inspected to classify according to the domain to which it belongs. This can be done by extracting the self-description part of the web page i.e. the meta tag and the title tag. After this the statistical values for the web page will be calculated. The determination of content genre plays useful role while determining the useful node of the web page. before determining the useful node in the web pages, it checks the context of the web page and identifies which threshold values to be calculated for the web page.

![Content Extraction Process](image)

In content genre determination module the web page is given as input and the contents of the meta tag are extracted to classify the website for the genre. If the website description matches the list of pre-classified genre of websites then the content extraction is done accordingly for that website. Example, if the website is of news domain then the content extraction has to be done only for the article in the website rather if the website is a shopping site too much removal of the links would render no useful information to the user.
RESULTS AND DISCUSSION

The system presents a hybrid model for content extraction from HTML documents. The proposed hybrid model is derived from two different models based on statistical features and formatting characteristics. The model works on Document Object Model (DOM) tree of the corresponding HTML document. It evaluates each tree node and calculates the associated statistical features like Deviation, Normalized Deviation, Link Density, and Normalized Link density to predict the significance of the node towards overall content provided by the document. Once the significance of the nodes is determined, then the usefulness of the node is predicted. Once significance of the nodes is determined, the formatting characteristics like fonts, styles, and the position of the nodes are evaluated to identify the nodes with similar formatting to compare it to the significant nodes. Such styles are used as a guide to understand the template of the document and serve as a backup mechanism to extract content from the nodes which have sometimes uncommon statistical values.

Calculation of Statistical Features:

The implementation has taken a sample page Information from the following Wikipedia page (http://en.wikipedia.org/wiki/Information) to calculate statistical features. The text Deviation ($D$) values obtained by using Eq (1) are shown in below Fig.2. Deviation will yield lower values for the nodes which have smaller quantities of text associated with them.

Fig. 2: Deviation in Text sizes for sample Information Page

The normalized deviation ($N$) is obtained from eq.(2) identifies the less informative nodes of DOM tree. The less informative nodes will yield very low values while the nodes which are used in rendering the most of the information have values nearer to 1.

Fig. 3: Normalized Deviation for sample information page
The normalization obtained by eq(2) (shown in Fig. 3) is also used for smoothing the deviation line graph shown in Fig. 2. The normalized deviation distinguishes main content areas, from the nodes like headers and advertisements, which are contributing less towards information being rendered by the page. However this does not suffice to distinguish them from the informative nodes of DOM tree that have less quantity of information as for example the titles of the columns in tables, headings. To make this distinction clear, the link densities \((L)\) by using Eq.3) are shown in Fig. 4. The Normalized Link Density \((NL)\) by using eq(4) they are as shown in Fig. 5.

Comparing the \(NL\) with \(N\) at any DOM node, Classifies that whether the DOM node contributes towards information or it is there for navigation. For the advertisements, headers or footers in any page, the link densities are usually high values. The system incorporates these statistical features extracted from any DOM node to classify if the node is of informative or non-informative nature. The performance of accuracy on wiki page of information is calculated by the amount of text present in actual page and the amount of text actually retrieved from that page. The performance measures are evaluated for first 15 nodes in page as shown in graph below in Fig. 6.
Conclusion:
The proposed system can now accept user input as URL in the browser, fetch the Web page and can construct the Document Object Model from that web page.

One among the drawbacks of the proposed system was its disability to detect the web page type or context of web page. Another drawback noticed was during the determination of useful nodes in statistical feature generation, wherein it tried to discard some relevant linkages as the weight was found below the threshold irrespective of determining whether discarded was relevant or not. The notion of enhancement surrounds these two drawbacks and looks on how by covering them it can increase the overall performance i.e. accuracy of the content extracted. These two drawbacks are overcome by classifying the content genre of the webpage and then extracting the content according to the threshold defined these leads in more accurate content extraction.

REFERENCES


Chakrabarti, S., 2003. Mining the Web : Discovering Knowledge from Hypertext Data . Morgan Kaufmann Publishers,


