A Survey of Web Information Extraction Tools

Noha Negm
Faculty of Science, Menoufia University (Egypt)

Passent ElKafrawy
Faculty of Science, Menoufia University (Egypt)

Abdel Badea Salem
Faculty of Computers and Information, Ain Shams University (Egypt)

ABSTRACT
The access to huge amount of information sources on the internet has been limited to browsing and searching due to the heterogeneity and the lack of structure of the web information sources. This has resulted in the need for automated Web Information Extraction (IE) tools that analyze the Web pages and harvest useful information from noisy content for any further analysis. The goal of this survey is to provide a comprehensive review of the major Web IE tools that used for Web text and based on Document Object Model for representing the web pages. This paper compares them in three dimensions: (1) the source of content extraction, (2) the techniques used, and (3) the features of the tools, moreover the advantages and disadvantages for each tool. Based on this survey, we can decide which suitable Web IE tool will be integrated in our future work in Web Text Mining.

General Terms
Knowledge Engineering

Keywords
Knowledge Engineering, Document Engineering, Information Extraction, Document Object Model, Web Documents

1. INTRODUCTION
With increasing development of the internet technologies, the web increases rapidly. Most of the web resources are in the form of Hypertext Markup Language (HTML) documents, which are viewed by web browsers. Therefore it becomes more and more popular. Therefore, the need of automated, flexible Web Information Extraction (IE) tools that extract information and knowledge from the Web pages and transfer it into a meaning and useful structures for further analysis will become a great necessity. There are a lot of approaches have been developed in the area of Web IE which concerns with how to harvest useful information for any further analysis from web pages. This type of web pages is typically generated by using a template and a set of text [1].

In this paper, we are concerned with the task of Web IE which differs from traditional IE task. Web IE aims at extracting text from online documents that are semi-structured and usually generated automatically by a server-side application program. While unstructured free texts that are written in natural language are the source of extraction in traditional IE. Consequently, traditional IE depends on the features of NLP techniques such as lexicons and grammars. In Web IE, usually machine learning and pattern mining techniques are applied to exploit the syntactical patterns or layout structures of the template-based documents.

Moreover, the Web page as a main source of data consists of many parts which are not equally important. Beside the main content, a Web page also comprises of noisy parts such as *advertisements, navigational bars on top and/or on the side, horizontal or vertical banner ads, boxes with links to related content, boxes containing images or animated ads, etc.* Although such information items are functionally useful for human viewers and necessary for the Web site owners, they often hamper automated information gathering, Web crawlers, and Web miners. So eliminating them will be a saving in storage, timing and indexing. Therefore, many approaches [2, 3, 4, 5, 6, 7, 8, 9, 10] to identify and extract main content are produced to alleviate this problem.

From time to time, many Web IE tools have been developed. If we wish to integrate one of them, then we would like to evaluate the existing tools to choose the one that fits our case better. One of the following three tasks should be performed: 1) study existing surveys that compare information extraction tools, 2) ask authors for these tools and then work on integrating them into other system, and 3) develop these tools to test them over the Web sites. At the end of any of the previous tasks, an IE tool is chosen and integrated, but the last two tasks are very high costing tasks. In this paper, we introduce a comprehensive review of different Web IE tools that have been used for Web text not data as the previous surveys. Based on this survey, we can choose the suitable Web IE tool that will be integrated in our future work in Web Text Mining.

The rest of the paper is organized as follows. Section 2 introduces the literature review. We make a study on different Web IE tools in Section 3. A comparative interpretation of the surveyed Web IE tools from the three dimensions is presented in Section 4. Finally, the conclusions are made in Section 5.

2. LITERATURE REVIEW
Extracting information and removing noise from web pages have been studied in many previous approaches. The majority of traditional approaches lead to eliminating the Webpage's inherent look-and-feel by increasing font size, removing images, disabling JavaScript, etc. The tools in [11, 12, 13] use hard coded techniques for certain common webpage designs as well as fixed "blacklists" of advertisers. The drawbacks of these tools are the possibility of producing inaccurate results if the software encounters a layout that it hasn't been programmed to handle. Instead of eliminate noise; there is an approach to reorganize it by Reformatting Content as in [14] which utilized their proprietary Small Screen Rendering technology that reformats Web pages to fit inside the screen width. The method of extracting content from "single-article" sources is presented in [15]. This approach works well for single-body documents, but destroys the structure of the HTML and doesn't produce good results for multi-body documents. The method of detecting the largest body of text on a simple Web page and classifying them as content is presented in [16, 17]. It works well with simple pages but in multi-body documents with random advertisement and image placement, it produces noisy or inaccurate results. The

methodology of the technique that uses structural analysis, contextual analysis, and summarization is presented in [18]. The first attempt to transform a web page into a hierarchy of individual content units called Semantic Textual Units is presented in [19, 20]. They depend on the analyzing of the syntactic features of an HTML documents in building STU. Moreover, they defined "accordion summarization" as a strategy where a page can be shrunk or expanded much like the instrument. A similar method to STUs for dividing a web page into a series of hierarchical "cards" that are placed into a "deck" is presented in [21]. This deck of cards is presented to the user one card at a time for easy browsing. The drawback of the deck-of-cards model is that it relies on splitting a page into tiny sections that can then be browsed as windows. But this means that it is up to the user to determine on which cards the actual contents are located. They also suggest a simple conversion of HTML content to WML (Wireless Markup Language), resulting in the removal of simple information such as images and bitmaps from the Web page so that scrolling is minimized for small displays.

Although the previous approaches do provide simpler means in which the content can be found, none of them solve the problem of automatically extracting just the content. Thus, these approaches limit analysis of Web pages. So that the researchers proposed a new technique that can remove the noise without destroying Web page layout, making more of a page's content viewable at once. The key is to work with the Document Object Model (DOM), rather than with raw HTML markup.

3. STUDY ON WEB IE TOOLS

The variety of information extraction tools motivated the existence of many surveys to compare these extraction tools such as [22, 23, 24]. They used different taxonomies for only how to extract information from web page by using Wrappers. However there are other tools used for this purpose such as *DOM Tree Exploration*, *Machine Learning*, and *Natural Language Processing*. Moreover, the comparison is done based on different points of views for researchers. While *wrappers* can provide excellent text extraction, they work only on specific pages, or sets of pages that share a common layout. Once the layout changes, the wrappers need to be updated as well. This brittleness unfortunately requires continuous supervision of wrapper-based approaches. In contrast to wrapper approaches, *Machine Learning techniques* can be used across sites, but again need to be re-trained as sites evolve. A third family of approaches applies *Natural Language Processing* techniques. These algorithms automatically extract named entities and parts of speech, thereby identifying page fragments that are most likely to be continuous, body prose. The disadvantage of NLP approaches for large databases is that the algorithms are often expensive. Their complexity thereby stands in the way of large scale analysis over sizable archives of pages [25].

3.1 Document Object Model

Document Object Model defines a common interface to a Web page. It provides logical access to the content, structure and style of a page through an ordered tree of element, text, comment and attributes nodes. Although each node in the DOM tree contains vital information, it is only in the element and text nodes where the actual content of a Web page can be found. Attributes and comments are not visible to the end-user when the page is rendered in a browser. Moreover, the visible content of a Web page lies in essence in the leaves of the DOM tree. The extraction of content can thus be formulated as a classification problem in which we seek to assign the right semantic label to each visible leaf node in the DOM tree. Figure 1 shows a segment of HTML codes and its corresponding DOM tree [29].

```
<BODY bgcolor=WHITE>
<TABLE width=800
height=200 >
……..
</TABLE>
<IMG src="image.gif"
width=800>
<TABLE bgcolor=RED>
……
</TABLE>
</BODY>
```

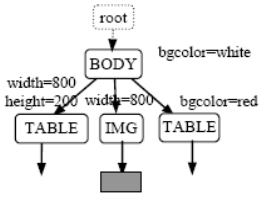

Fig 1: A DOM tree example [29]

Using DOM tree as a source of extraction instead of web page has many benefits such as: by parsing a Web page into a DOM tree, more control can be achieved while extracting content since. Furthermore, we can not only extract information from large logical units but can also manipulate smaller units such as specific links within the structure of the DOM tree. In addition, DOM trees are highly editable and can be easily used to reconstruct a complete Web page.

3.2 Three Dimensions for Comparing Web IE Tools

There are three dimensions to be used in the comparison: (1) The first dimension determines the source of content extraction, (2) The second dimension compares the techniques used in different Web IE tools, and (3) The third dimension determines the feature of each tool. In this section we describe each of these dimensions that can be used as criteria for comparing and evaluating IE tools.

1) The Source of Extraction. The criteria of the first dimension determine the source type of the input file. The input file of an IE task may be structured, semi-structured or free-text. The definition of these terms varies across research domains. In [26] they considered free-texts e.g. news article are unstructured, postings on news-group, medical records are semi-structured, while HTML pages are structured. However, in [27], the information stored in databases is known as structured data; XML documents are semi-structured data, while web pages in HTML are unstructured because there is very limited indication of the type of data. We agree with the viewpoint in [23] since XML documents are considered as structured, free texts are unstructured and the HTML pages on the Web are considered as semi-structured. For many Web IE tasks, the source of extraction may be multiple Web pages for different web sites or a set of web pages from the same web sites.

2) The Techniques Used. The criteria of the second dimension classify Web IE tools based on the techniques used. The web information extraction tools can be divided into two classes: tag dependent and tag independent. Tag dependent technique which concerns with all tabular documents that contain a <Table> tag, while tag independent technique don't dependent on the tags.

3) The Features Tools. The criteria of the third dimension determine the feature of each tool. Firstly, measure the degree of automation for each tool. It evaluates both the effort made

by the user for the training process and the necessity to port Web IE tool across different domains. In some Web IE tools, there are many phases that require the user to accomplish it such as collecting training pages, labeling training examples, generalizing extraction rules, extracting the relevant data, and outputting the result in an appropriate format these tools are called Semi-automatically. On the other hand, some IE systems do not require the collected training examples. This brings up a major difference in automation these tools called fully-automatically. Secondly, determine the source of extraction for each tool. In this survey we select all Web IE tools that based on the DOM tree as the source of extraction.

3.3 Web IE Tools

In this section, we will survey the majority of the Web IE tools during the period from (2003) to (2010) that based on Document Object Model technique.

3.3.1 Crunch (1.0, 2.0)

Crunch 1.0 framework is a content extraction program developed by Gupta et al [3]. Instead of using raw HTML text, it uses the DOM tree representation of a web document. Receiving input of a HTML page, Crunch will parse the HTML string, construct the DOM, traverse the nodes recursively and filter out the non-informative content behind. Crunch 1.0 is a one-pass system that extracted content by running a series on filters one after the other, i.e., the selected filters just ran sequentially on the output produced by the previous filters. Each of the filters can be turned on or off and customized to certain degree. Generally there are two sets of filter in both versions of Crunch. The first set simply ignores specific HTML tag such as styles, links, and images. The second set consists of the advertisement remover, the link list remover, the empty table remover, and the removed link retainer. After performing a series of filtering pipeline the final output of Crunch can be customized as a transformed HTML document or plain text format.

To overcome the pitfalls of version 1.0 where occasionally pages returned null outputs after passing through Crunch, e.g., link heavy pages like www.msn.com, the newer version Crunch 2.0 developed [28]. It is a multi-pass system since multiple copies of a webpage kept in memory and a filter checked for the optimal copy to work on. In Crunch 2.0, new filters that allow the user to control the font size and word wrapping of the output, and heuristic functions guiding the multi-pass processor to evaluate the acceptability of the output as each filter pass edits the DOM tree are added.

3.3.2 InfoDiscoverer Algorithm

It is [2] proposed to efficiently and automatically discover intra-page redundancy and extract informative contents of a page. The researchers concentrated on HTML documents with <TABLE> tags. Based on HTML tag <TABLE>, a page is partitioned into several content blocks. Based on DOM, a coarse tree structure is obtained by parsing an HTML page based on <TABLE>. Each internal node shows a content block containing one or more content strings as its leaf nodes. After parsing a page into content blocks, features of each block are extracted. The features mean the meaningful keywords. After extracting features entropy value of a feature is calculated according to the weight distribution of features appearing in a page cluster. Next step is calculation of entropy value of a content block. It is given by summation of its features entropies. The entropy of a content block is the average of all entropy values in that block. By using this, a content block is identified as informative or redundant. If the entropy value is higher than a threshold or close to 1 then content block is redundant as most of the block features appear in every page. If it is less than a threshold then the content block is informative as features o the page are distinguishable. The algorithm is not applicable to general web pages; moreover it is evaluated for only Chinese pages published by news web sites not for English web pages.

3.3.3 Site Style Trees (SST)

It is [29] considered non-content blocks as local noise in the web page. They used a tree structure called Site Style Tree (SST), to capture common presentation styles and actual contents of the pages in the given web site. A Style Tree can be built for the site by sampling the pages of the site. As known each HTML page corresponds to a DOM tree where tags are internal nodes and the detailed texts, images or hyperlinks are the leaf nodes. A DOM tree can represent the presentation style of a single HTML page, but it is difficult to study the overall presentation style and content of a set of HTML pages. It is difficult to clean pages based on individual DOM trees. So a tree structure known as Style Tree is used for this purpose. Two importance values, presentation importance and content importance, are used to find importance of an element node. The presentation importance is used to detect noises with regular presentation styles while content importance is used to detect those main contents of the pages that may be represented in similar presentation styles. Hence, the importance of an element node is given by combining its presentation importance and content importance.

3.3.4 ContentExtractor, FeatureExtractor, K-FeatureExtractor, L-Extractor Algorithms

It is [30] proposed based on the same basic concept used by Lin [2], that a <TABLE> tag is used to design maximum Web pages. Unlike Lin they make use of some other HTML tags also while designing the algorithms. In a table occurring in a Web page, each cell is considered as a block. Where tables are not available, blocks can be identified by partitioning a Web page into sections that are coherent and that have specific functions. Similar blocks across different Web pages obtained from different Web sites can be identified using these algorithms. An added advantage of identifying blocks in Web pages is that if the user does not require the non content blocks or requires only a few non content blocks, the rest of the blocks can be deleted. This contraction is useful in indexed and stored since eliminating the non content blocks result in significant savings with respect to storage cache and indexing.

The algorithms first partition the Web page into blocks based on heuristics (different HTML tags). Secondly, the algorithms classify each block as either a content block or a non content block. While the algorithm decides whether a block *B* is content or not, it also compares *B* with stored blocks to determine whether *B* is similar to a stored block. *ContentExtractor* identifies non-content blocks on the appearance of the same block in multiple web pages. *FeatureExtractor* is based on this characterization and uses heuristics based on the occurrence of certain features to identify content blocks. *K-FeatureExtractor* is a special modification of *FeatureExtractor* which performs better in a wide variety of web pages, and *L-Extractor* uses various block features and train a Support Vector Model (SVM) based classifier to identify an informative block vs. a non informative block.

3.3.5 HTML2RSS System [5].

Rich Site Summary (RSS) is an XML-based format that allows web developers to describe and syndicate web site content. A lot of famous web sites have provided RSS for users. However, making RSS files manually is boring, and so far, most sites have not provided such a service. RSS is now becoming an efficient way to distribute information for publishers, and obtain information for plain readers. Hua Geng et al [5] proposed a HTML2RSS system, using two new algorithms SAMR and ATP, to extract content from HTML web pages based on DOM structure, and generate RSS files automatically with the extracted content as a substitute of the HTML page. Because of the flexibility of HTML syntax, a lot of news web pages do not strictly obey the W3CHTML specification, which may cause mistakes in DOM tree structure. So that the first step in the system is using the HTML parser and after that a DOM tree is generated. In SAMR algorithm, the Mapping Rule is created and used to extract information for creating the RSS file. On news web pages, a news item is often published with the corresponding release time. This feature is a prominent and useful clue for locating and extracting the target news items. With this characteristic, the ATP algorithm is designed. Since the formats of data and time are simple and limited, it is easy to construct a database called Time Pattern DB for storing and managing time patterns, which are abstract representations of time formats.

3.3.6 CoreEx Algorithm

It is [6] developed a heuristic technique for automatically extracting the main article from online news site web pages. CoreEx used a DOM tree representation of each page, where every node in the tree represents an HTML node in the page. Moreover, the amount of text and the number of links in every node are analyzed and they used a heuristic measure to determine the node (or a set of nodes) most likely to contain the main content. For every node in the DOM tree, two counts are maintained *textCnt* that holds the number of words contained in the node and *linkCnt* that holds the number of links in or below the node. A node's score is a function of its *textCnt* and linkCnt. If two nodes reach identical scores, the node higher in the DOM tree is selected.

3.3.7 Node Information Threshold (NIT) Algorithm

It is [31] proposed a new method for information extraction from web pages. The method is based on statistical analysis of web page content and intended mainly for text corpus making. The NIT method is an automatic statistical based algorithm using the web page structure for information extraction. It transforms the source document into the hierarchical structure DOM tree. Each DOM node represents one web page element. The NIT method is based on detection of most useful nodes in the DOM tree. The useful nodes are included into the extraction result and they represent an ideal plain text extract in the best case.

3.3.8 Content Seeker (CS).

Generally most of the existing content extraction approaches that used heuristic rules introduced only one main feature to distinguish main content from noisy information. Combining of different features usually can lead to better content extraction as different kinds of web pages have different characteristics. Samuel Louvan [7] proposed a new hybrid approach that consist of Machine Learning and developed heuristic approaches namely Largest Block String (LBS), String Length Smoothing (SLS), and Table Pattern (TP). With the Machine Learning approach, many kinds of features can be used and the learning algorithm may learn the parameter automatically. Moreover, by using this approach different learning results can be applied for different types of websites**.**

3.3.9 Web article and image extraction system.

Most existing approaches such as [2, 3, 22, 29] rely on prior knowledge of website specific templates and hard-crafted rules specific to websites for extracting of relevant content. Parag et al [32] proposed a generic approach that does not require prior knowledge of website templates. It is a combination of HTML DOM analysis and Natural Language Processing (NLP) techniques for automated extractions of main article with associated images from web pages. The first stage of the system is to extract the article body, which is followed by NLP analysis to fine the relevant article images. In the next step, content sub-blocks in the article block are further analyzed to eliminate easy-to-identify unwanted blocks such as lists of links. The remaining content should be article text body and any images around the article block. Finally, the semantic similarity algorithm based on NLP tries to associate relevant images with main text content based on captions around images. The semantic similarity algorithm also eliminates unrelated images such as advertisements.

3.3.10 Visual Clustering Extractor algorithm (VCE)

Most of the proposed algorithms are tag dependent means they could only look for primary content among specific tags such as <TABLE> or <DIV>. Mohsen Asfia et al [33] introduced a new algorithm which simulates a web page user visit and how the user finds the main content block position in the page. The proposed method is tag independent and has two phases to accomplish the extraction job. First, it transforms input DOM tree obtained from input HTML detailed web page into a block tree based on their visual representation. Second, every node in the DOM structure will have specification vector. It traverses the obtained small block tree to find main block having dominant computed value in comparison with other block nodes based on its specification vector values.

4. COMPARATIVE INTERPRETATION OF WEB IE TOOLS

As we show there are many researchers have developed various tools for text extraction from Web pages. This is the first study to compare such tools based on the DOM tree as the main source of extraction. In this section, we use the criteria of the three dimensions suggested in section (3.1) to compare the surveyed Web IE tools. We collected the final comparison between all tools in the Table 1.

4.1 Source of Extraction -Based Comparison

In this section, we contrast among the capabilities of the surveyed Web IE systems to support various IE tasks. The features in this dimension include input variation, such as Web pages with the same structure or the structure of them is completely different, and output variation such as HTML documents or plain text, or the informative content block as shown in Table 1. Some of the Web IE tools created their dataset as collection of Web pages from different Web sites such as *Crunch* and *CoreEx*. Others tools created their dataset as a set of Web pages from the same Web site especially News

Web sites such as *K-FE* and *VCE* or Commercial Web sites such as *SST*.

4.2 Technique-Based Comparison

In this survey, we use the DOM tree representation as the source of extraction since we can obtain many kinds of features by accessing the DOM nodes. For DOM-based methods, usually the existing methods perform Web page segmentation prior to content extraction. The Web page segmentation process is needed because there are dozens of DOM nodes in a single Web page and we need to focus on DOM nodes in certain granularity. After parsing a Web page into a DOM tree, we observed that as shown in Table 2 the tools are divided into two categories.

The first category based on tag dependent, means they could only look for primary content among specific tags such as <TABLE> tag like *LH* algorithm or <DIV>tag like *Web article and image extraction* system. The second category is tag independent, means they used different techniques to locate the main content as shown in Table 1. For heuristic rules, most content extraction approaches introduced one main feature to distinguish main content from noisy content like *CoreEx* and *NIT*, or using learning algorithms to combine many kinds of different features like *CS*, or applying series of filters like *CRUNCH*. Table 2 summarized each Web IE systems from the perspective of the underlying techniques used.

4.3 Features-Based Comparison

In this section, we compare and evaluate web IE systems from the automation degree prospective. The results are shown in Table 1. From Table 1, we noticed that there is no manual IE systems most of them are semi-or fully automatically. Semi-automatically means the systems require little effort of user for collecting training pages or labeling training examples such as *HTML2RSS* System and *CS*. Others systems are fully automatically that are not require any effort from users in any stage of the system such as *K-FE* and *CoreEx* algorithm.

Table 2. Analysis for each Web IE tools based on input task and techniques used

TOOLS	Web Page	Web Site	Tag dependent	Tag independent
CRUNCH 1.0, 2.0	☑			☑
LH		☑	☑	
SST		☑		☑
CE		☑	☑	
FE		☑	☑	
K-FE		☑	☑	
LE		☑	☑	
HTML2RSS	☑			☑
CoreEx	☑			☑
NIT	☑			☑
CS	☑			☑
article and image	☑		☑	
VCE		☑		☑

4.4 Overall Comparison

Although we have compared various Web IE systems from three dimensions, we summarize and present all advantages and disadvantages of these Web IE systems in Table 3. Moreover, another comparison between the above tools is made based on their experimental results in [4, 33]. All tools used the same dataset from the News domain for example, CNN, BBC, and Yahoo ….etc. The three basic measures used for evaluating the performance of the tools are *Precision*, *Recall* and *F-measure*. Table 4 shows the Precision, Recall and F-measure values for each tool at some news websites.

Table 4. Block level comparison between some Web IE tools [4, 33]

The Tools	News Web sites								
	Precision			*Recall*			*F-measure*		
	CNN	*BBC*	*YAHOO*	*CNN*	*BBC*	*YAHOO*	*CNN*	*BBC*	*YAHOO*
LH	0.856	0.834	0.730	1	0.99	1	0.922	0.905	0.84
CE	0.977	0.968	0.967	1	1	1	0.988	0.983	0.98
FE	0.98	1	1	0.98	1	0.94	0.97	0.99	0.972
K-FE	0.98	1	1	0.98	1	0.95	0.98	1	0.974
VCE	1	0.98	1	1	1	1	1	0.98	1

From Figures 2, 3 and 4, it is clear that the tools CE, FE, K-FE and VCE outperform LH in all news sites. Moreover, the FE and K-FE have the same values in most of the news sites. Furthermore, VCE has higher Precision, Recall and F-measure in comparable to K-FE. Table 5 summarized the final comparison between the best two K-FE and VCE tools. It noticed that the *K*-FE algorithm gives excellent Recall, Precision, and F-measure in addition to efficient runtime in all types of Web sites especially for shopping and Review sites, Chat forums that all contain multiple blocks of important textual information [4]. In the News Web sites, the performance of VCE algorithm is higher in Recall, Precision, and F-measure than *K*-FE algorithm [33].

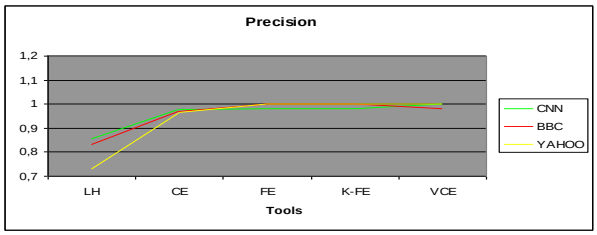

Fig 2: Precision-Based comparison in the three Websites

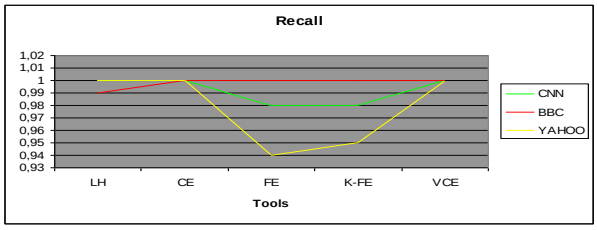

Fig 3: Recall-Based comparison in the three Websites

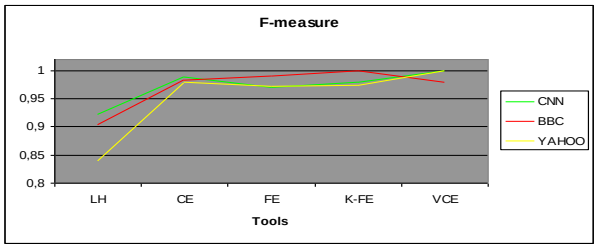

Fig 4: F-measure-Based comparison in the three Websites

Table 5. Overall comparison between K-FE and VCE

	K-FE Algorithm	VCE Algorithm
Technique Used	Tag dependent	Tag independent
Domain	Different domains	Only one domain
Data sets	Different Websites include News, Shopping, Opining Posting: ABC, BB, BBC, CBS, CNN, FOX, FOX23, IE, IT, MSNBC, YAHOO, Shopping, Amazon, Barnes and Noble, Epinion	Only News Web sites that are used in K-FE algorithm such as : ABC, BBC, CBS, CNN, FOX, FOX23, MSNBC, YAHOO
Advantages of Evaluation	- Give excellent *Recall*, *Precision*, and *F-measure* and runtime efficiency in all types of Web sites especially for Shopping and Review sites., Chat forums that all contain multiple blocks of important textual information. - Above all, do not use any manual input and require no complex machine learning process	- The performance of the algorithm is similar to the *K-FE* algorithm and in some cases gives higher *Recall*, *Precision* and *F-measure* than it in the some News Web sites - Doesn't have any learning phases and could find informative content on any random input detailed Web page.
Main process	- Input - Segmentation process (based on heuristics) - Classification process - Output	- Input - Constructing Block Tree - Find the Main Block based on BT -Output
Degree of Automation	Fully Automatically	
The output	Text content of all informative blocks	

5. CONCLUSIONS

In this paper, we survey the major Web IE tools in the literature and compare them in three dimensions: the source of extraction, the features, and the techniques used. The criteria of the first dimension include input variation, such as single web page, multiple web pages, and output variation such as HTML documents or plain text. The criteria of the second dimension measure the performance of IE systems. The criteria of the third dimension measure the degree of automation for IE systems. All surveys IE tools used the Document Object Model for representing the web pages. Moreover, we studied the advantages and disadvantages of all tools to enable us to choose the appropriate tool that will be used in our work. Based on this survey we will integrate the VCE tool for extracting informative content from the News domain in our future work.

Table 1. Analysis for each Web IE tools based on the three dimensions

The Tool	Year	Input domain	Features Tool	Techniques used	Output result
LH Algorithm	2002	Web pages	- DOM - Fully automatic	- Parsing a page into content blocks based on <TABLE> tags. - An Entropy based method used to calculate the entropy value of a content block.	Informative content blocks
Crunch 1.0	2003	Web pages	- DOM - Semi-automatic	- HTML parser that creates a DOM tree representation of the web page. - A series of different filtering techniques	HTML or plain text (article text-body only)
SST	2003	Web site	- DOM - Fully automatic	- DOM tree for each page prepared. - All DOM trees merged in a top down manner to form a style tree.	Cleaned Web pages with only meaningful nodes
Crunch 2.0	2005	Web pages	- DOM - Semi-automatic	- HTML parser to create a DOM tree - Using a series of different filtering techniques - Using new filters to control the font size and word wrapping of the output	HTML document or plain text format
CE Algorithm	2005	News Web site	- DOM - Fully automatic	- Partition each page into blocks. - The inverse block document frequency used to eliminate redundant blocks.	Primary content blocks
FE Algorithm	2005	Web pages	- DOM - Fully automatic	- Partition the page into blocks. - Calculates the probability of individual blocks for the desired feature. - Takes those blocks in a single winner set.	Text contents of the informative block with the highest probability
K-FE Algorithm	2005	News Web site	- DOM - Fully automatic	- It uses an adaptive *K*-means clustering algorithm to retrieve multiple winners. - After the clustering is done, the high probability clusters are taken.	Text contents of all informative blocks
LE Algorithm	2005	News Web site	- DOM - Fully automatic	- It uses various block features. - An SVM based classifier is used to identify primary content blocks.	Text contents of all informative blocks
HTML2RSS	2007	Web pages	- DOM - Semi-automatic	- HTML parser to create a DOM tree - The SAMR algorithm used to create the Mapping Rule and extract information for creating the RSS file. - The ATP algorithm used for locating and extracting the target news items.	RSS file with extracted content
CoreEx Algorithm	2008	Web pages	- DOM - Fully automatic	- HTML parser to create a DOM tree -The amount of text (textCnt) and the number of links (linkCnt) in every node analyzed. - A heuristic measure (Weighted Scoring function) used to determine the node most likely to contain the main content.	Article nodes that contain the content
NIT Algorithm	2008	Web pages	- DOM - Fully automatic	- Creates a DOM tree representation - Apply the I(N) function that evaluates the information measure of the node *N*	Useful plain-text and associated metadata

CS	2009	Web pages	- DOM - Semi-automatic	- Segment Classification by using DOM tree - Content Classification as main content or noisy content - Heuristic rules to filter the DOM nodes	Text strings of the main content
article & image system	2009	Web pages	- DOM - Fully automatic	- HTML parser to create a DOM tree - Content object filter to eliminate unwanted blocks - Applying the semantic similarity algorithm that relies on NLP to extract relevant image	Article text body with associated images
VCE Algorithm	2010	Web site	- DOM - Fully automatic	- HTML parser to create a DOM tree - Use *GetBlockTree* algorithm to return block tree and general parameters - Use *FindMainBlock* algorithm to return the main block	The informative content block

Table 3. Advantages and disadvantages for each surveyed IE tools

The Tool	**Advantages**	**Disadvantages**
LH Algorithm	- It is efficiently and automatically to discover intra-page redundancy (advertisements, a search box, the copyright, etc.) and extract informative contents of a page (news content)	- The system knows a priori how a web page can be partitioned into coherent content blocks. - The system knows a priori which blocks are the same blocks in different web pages. - It works at the feature level instead of the block level
Crunch 1.0	- It not only finds the content but also eliminates the non-content. - The algorithm performed well on pages with large blocks of text such as news articles and mid-size to long informational passages.	- Most of the performance drop from using the proxy originates from the proxy's need to download the entire page before sending it to the client - It don't get better results with the link-heavy pages
SST	- It is an efficient technique to study the overall presentation style and content of a set of HTML pages.	- Style tree is an efficient technique to detect content blocks but constructing these style trees is a complex task.
Crunch 2.0	- It prevents link-heavy pages like www.msn.com from returning blank pages as output. - It doesn't technically find the content but instead eliminates likely non-content.	- It does not allow re-training process as the heuristic rules and threshold values are fixed. - Crunch cannot filter non-HTML content
CE Algorithm	- It produces excellent precision and recall values and runtime efficiency - It detects redundant blocks based on the occurrence of the same block across multiple Web pages.	- If there are web pages whose elements have the same style but different contents which are non content blocks, then the algorithm would not be able to detect that.
FE Algorithm	- It is give better results in comparison with other algorithms in precision and recall values. - It do not use any manual input and require no complex machine learning process	- In the News Web sites, the performance of FE algorithm is lower in Recall, Precision, and F-measure
K-FE Algorithm	- Unlike the ContentExtractor, it does not depend on multiple web pages but depends on the feature set and the chosen feature for output.	- It shows poor precision and recall as it produces only one text block with highest probability, while other important blocks are not retrieved
LE Algorithm	- It produces fairly good accuracy - Combining the block partitioning algorithm with an SVM works with high efficiency.	- It can't extract the main content on any random input detailed Web page.
HTML2RSS	- It extracts content from HTML web pages based on DOM structure, and generates RSS files automatically with the extracted content as a substitute of the HTML page.	- Although SAMR algorithm is more accurate and can be applied to almost any news web page, it needs manual work during creating Mapping Rules, which will take much more time than RSS generating.
CoreEx Algorithm	- It works well for any structure where content nodes differ from the non-content nodes significantly in the amount of text and links they contain - The method is site-independent and does not use any language based features.	- The approach developed only for news articles, it not gives a good performance in any other types of pages such as blogs, product pages, reviews and facebook pages. - The approach doesn't handle the article titles and image captions.

NIT Algorithm	- It is an automatic statistical based algorithm using the web page structure for information extraction. - It is appropriate for large corpus building from web sources because the setup depends only on a single parameter.	- The pitfall of this method is the inclusion of the footer into extract, which decreases the precision especially for short articles
CS	- It is based on the combination of machine learning and heuristic rules - It is extract contents from different kinds of web pages that have different characteristics.	- The approach requires labeled training data
article & image system	- It does not require prior knowledge of website templates. - It does not focus on extracting the article text body only but also extracting associated images	- It is a difficult task to accurately identify informative images without any semantic information. - It can't extract non-HTML content.
VCE Algorithm	- It is not dependant on any tag type and it just has an iteration to block its input page while - It doesn't have any learning phase. - It could find informative content on any random input detailed web page.	

6. REFERENCES

[1] H. Sleiman, "Information extraction from the World Wide Web", Actas de los Talleres de las Jornadas de Ingeniería del Softwarey Bases de Datos, 2009.

[2] Lin, S. and Ho, J. 2002. Discovering informative content blocks from web documents. In Proceeding of the 8th International *KD* and *DM* Conference.

[3] Gupta, S., Kaiser, G., Neistadt, D. and Grimm, P. 2003. Dom-based content extraction of html documents. In Proceeding of the 12th International Conference on World Wide Web.

[4] Debnath, S., Mitra, P. and Giles, C. 2005. Automatic Extraction of Informative Blocks from WebPages. In Proceeding of the 20th Annual ACM *SAC'05*.

[5] H. Geng, Q. Gao, and J. Pan, "Extracting Content for News Web Pages based on DOM", International Journal of Computer Science and Network Security, 2007.

[6] Gibson, J., Wellner, B. and Lubar, S. 2008. CoreEx: content extraction from online news articles. In Proceeding of the 17th ACM *IKM* Conference.

[7] Louvan, S. 2009 Extracting the Main Content from HTML Documents. [Online]. Available: http://wwwis.win.tue.nl/bnaic2009/papers/bnaic2009_paper_113.pdf

[8] Guo, Y., Tang, H., Song, L., Wang, Y. and Ding, G. 2010. ECON: An Approach to Extract Content from Web News Page. In Proceeding of the 12th International Asia-Pacific Web Conference.

[9] Spengler, A. and Gallinari, P. 2009. Learning to Extract Content from News WebPages. In Proceeding of the International Conference on Advanced Information Networking and Applications Workshops.

[10] Pasternack, J. and Roth, D. 2009. Extracting Article Text from the Web with Maximum Subsequence Segmentation. In Proceeding of the International World Wide Web Conference Committee (IW3C2).

[11] Wpar homepage on Sourceforge 2012. [Online]Available: http://sourceforge.net/projects/wpar

[12] The Webwiper website 2012. [Online]. Available: http://www.webwiper.com

[13] The junkbusters website 2012. [Online]. Available: http://www.junkbusters.com

[14] The Opera website 2012. [Online]. Available: http://www.opera.com

[15] Finn, A., Kushmerick, N. and Smyth, B. 2001. Fact or fiction: Content classification for digital libraries. In Joint DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries.

[16] McKeown, K., Barzilay, R., Evans, D., and Hatzivassiloglou, V., 2001. Columbia Multi-document Summarization: Approach and Evaluation. In Proceeding of the Document Understanding Conference

[17] Wacholder, N., Evans, D. and Klavans, J. 2001. Automatic Identification and Organization of Index Terms for Interactive Browsing. In Proceeding of the Joint Conference on Digital Libraries '01.

[18] Rahman, A., Alam, H. and Hartono, R. 2001. Content Extraction from HTML Documents. In 1st International Workshop on Web Document Analysis WDA.

[19] Buyukkokten, O., Molina, H. and Paepcke, A. 2001. Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones. In Proceeding of the Conference on Human Factors in Computing Systems.

[20] Buyukkokten, O., Molina, H. and Paepcke, A. 2001. Text Summarization for Web Browsing on Handheld Devices. In Proceeding of 10th International World-Wide Web Conference.

[21] Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S. and Laakko, T. 2000. Two Approaches to Bringing Internet Services to WAP Devices. In Proceeding of 9th International World-Wide Web Conference.

[22] laender, A., Ribeiro-Neto, B., Silva, A. and Teixeira, J. 2002. A brief survey of web data extraction tools. In Proceeding of SIGMOD Conference.

[23] C. Chang, M. Kayed, M. Girgis, and K. Shaalan, "A Survey of Web Information Extraction Systems", Journal of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2006.

[24] Fiumara, G. 2007. Automated Information Extraction from Web Sources: a Survey. In Proceeding of the 3rd International Conference on Communities and Technology.

[25] Liu, B. 2011 Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Second Edition. Springer-Verlag Berlin Heidelberg.

[26] Soderland, S. 1997. Learning to extract text-based information from the World Wide Web. In Proceeding of 3rd International Conference on Knowledge Discovery and Data Mining (KDD).

[27] Elmasri, R. and Navathe, S. 2003. Fundamentals of Database Systems. In Proceeding of the 4th Edition Addison Wesley.

[28] Gupta, S., Kaiser, G., Neistadt, D., Chiang, M., Starren, J. and Grimm, P. 2005. Automating Content Extraction of HTML Documents. Technical Report. University of Columbia at Computer Science.

[29] Yi, L., Liu, B. and Li, X. 2003. Eliminating noisy information in web pages for data mining. In Proceeding of the 9th ACM SIGKDD International Conference.

[30] S. Debnath, P. Mitra, N. Pal, and C. Giles, "Automatic Identification of Informative Sections of Web-pages", In Journal IEEE Transactions on Knowledge and Data Engineering, 2005.

[31] Toman, M. 2008. Comparison of Approaches for Information Extraction from the Web. In Proceeding of the 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint. Slovenia.

[32] Joshi, P., Liu, S. 2009. Web Document Text and Images Extraction using DOM Analysis and Natural Language Processing. In Proceeding of the 9th ACM *SDE* Conference.

[33] M. Asfia, M. Pedram and A. Rahmani, "Main Content Extraction from Detailed Web Pages", In Proceeding of International Journal of Computer Applications, 2010.