VisHue: Web Page Segmentation for an Improved Query Interface for MedlinePlus Medical Encyclopedia

Aastha Madaan, Wanming Chu, and Subhash Bhalla

University of Aizu, Aizu-Wakamatsu Shi,
Fukushima-ken, Japan 965-8580
{d8131102,w-chu,bhalla}@u-aizu.ac.jp

Abstract. World Wide Web has become the largest source of information. Consequently web based information retrieval, information extraction; automatic page adaptation and querying deep-web are gaining importance. The need for information retrieval applications is increasing. To address the problems of the ever expanding information over the internet, traditional information retrieval techniques have been applied. Such techniques are sometimes time consuming, and laborious, and the results obtained may be unsatisfactory. This study is an attempt to query web pages like MedlinePlus medical encyclopedia by segmenting the web pages. It summarizes the existing approaches for web page segmentation from the perspective of "structure realization for improved querying" on the web. It proposes a new algorithm *VisHue* for web page segmentation based on visual cues and heuristics and further uses the hierarchical structure generated by it to develop the Query by Segment or Tag (QBT) query interface. This interface is close to the end-user and exploits the relationships among the various content groups within a web page. Such an improved query-interface enables the user to perform in-depth querying. It is a step beyond the page-level search.

Keywords: Web page segmentation, hierarchical structure, advanced querying.

1 Introduction

The World Wide Web has become the most important source of information in the world. The family of algorithms for web focused information retrieval is growing. This is achieved by segmenting the web page, classifying resulting segments. Most information retrieval systems on the Web consider web pages as the smallest and undividable units, but a web page as a whole may not be appropriate to represent a single semantic. Some basic understanding of the structure and the semantics of web-pages could improve people's browsing and searching experience [22]. A web page usually contains various contents such as navigation, decoration, interaction and contact information, which are not related to the topic of the web-page. Furthermore, a web page often contains multiple topics that are not necessarily relevant to each other. Therefore, detecting the semantic content structure of a web page could potentially improve the performance of web information retrieval [1].

S. Kikuchi et al. (Eds.): DNIS 2011, LNCS 7108, pp. 89–108, 2011.

There are a large number of applications which make use of web page segmentation algorithms such as link analysis, topic distillation, focused crawling, improved querying, information accessing, overcoming the limitations of browsing and keyword searching, building wrappers to structure the web data. Such applications exploit the semantic structure within the web page. Furthermore, an acquisition, detection and analysis on web contents are paid more and more attention, web page segmentation algorithms are becoming an important part of them.

There has been plenty of work in this area. Some approaches worked on automating the web page, while some on the learning based splitting of the web page. Kai et al. [1] gave an algorithm that used rule based heuristics to segment the visual layout of a webpage. On the flip side, Kao et al. [13] gave an algorithm for webpage segmentation that relies on content based features. Other notable works used DOM node properties to find webpage segments. While Chakrabarti et al. [12], used template based segmentation for enhanced topic distillation. Chakrabarti et al. [11] also gave an algorithm based on isotonic regression whose by-product is a segmentation of a webpage into informative and non-informative parts [6].

The main aim of this study is to suggest the best approach for web page segmentation algorithm that can generate a hierarchical structure of the web page and improve the queryingfor websites that have pages like web-documents such as the medical encyclopedia entries [14], [15], [16], [17]. The content under these web pages is confined under a main node. In section 2, we introduce the concept of "web-page segmentation" its need and evolution of various approaches for it and explain their categorization.Section 3 presents the characteristics of a good web page segmentation algorithm and describe the features of each of the algorithms their strengths and weaknesses. We draw a detailed comparison of all the existing approaches w.r.t. facilitation in hierarchical structure generation. Among these approaches we mention our on-going work on a new visual cues and heuristic rules-based approach for webpage segmentation*VisHue*. In Section 4 we present the new query interface QBT (Query by Tag or Segment) based on the hierarchical structure constructed by *VisHue* and a qualitative and quantitative analysis of its efficiency in comparison with the keyword search. Section 5 presents the design and scope of improvements of this interface. Section 6 gives the summary and conclusions.

2 Background: Web Page Segmentation

As the amount of information and services available via the web increases, the use of web for accessing information for diverse activities such asshopping and communication is increasing. The changes have resulted in a more sophisticated presentation of content on the web. A web page typically displays a number of different messages to the user, which are usually visually distinct. For example, a web page might contain advertisements and links to other relevant pages in addition to the main content of the page. Thus, an application that intends to re-use content on the web, such as a search engine or a web-to-print application needs to identify the regions of the page that contain distinct information [10]. The presentation of a web page involves placing different pieces of information on it — each serving a different purpose to the end-user — in a manner that appears coherent to users who browse the webpage. These pieces have carefully-placed visual and other clues that cause most users to subconsciously segment the browser-rendered page into semantic regions, each with a different purpose and content [6].

Thus, segmentation of a web page can be defined as dividing a web page into structural blocks, each block may or may not contain templates or may be part of a template. Further, in a segmentation process an area that does not contain templates may be divided into several blocks [7]. It demarcates informative and non-informative content on a webpage; and also discriminates between different types of information. It is very useful in web ranking and web data mining applications. For instance, in a multiword query whose terms match across different segments in a page; this information can be useful in adjusting the relevance of the page to the query [8]. Also the user can be provided with an improved query interface where he can query the semantic groups individually.

2.1 Hierarchical Structure Generation

The hierarchical or logical structure of a document plays an important role in many applications. For example, work presented in [1] exploits the hierarchal structure of a document to carry out anaphora resolution. In [2], the logical structure is used to segment a web document and perform passage retrieval. Other applications that can make use of hierarchical structure include browsers designed for cell phones, PDAs and PCs with non-PC terminals as well as text summarization and data mining applications. However, the hierarchical structure of web documents is not always explicitly represented. Many web designers prefer to use their own styles to represent headings than to use the html heading tags meant to convey a document's logical structure. This limitation can be overcome by a heading detection algorithm and a level detection algorithm through which a document's hierarchical structure can be extracted[18].

Constructing a query interface using the hierarchical structure of the web page is beneficial as it can exploit the various relationships that exist amongst the various content groups and also has additional advantages like:

- In the same domain, there might be a case where important website query interfaces are required to be integrated, to create a unified query interface, if each of these interfaces is generated using a hierarchical structure of the web page, then it is easy to map the attributes.
- It also provides better query interface matching [19].
- Such interfaces are more close to the user's understanding and are qualitatively better than those generated by sources having a flat representation.
- The fields in such an interface are organized in a better manner with appropriate labels.

2.2 Web Segmentation Approaches

Until recently, it has been possible to identify distinct regions and components from the HTML code that generated a web page. The recent trend towards dynamic web technologies implies that the HTML no longer contains sufficient information on the contents of the page. Sometimes it contains almost no information, e.g., in the case of flash presentations where there is no content in the HTML DOM. Such pages however remain perfectly understandable to the user. Hence, the DOM-based segmentation became obsolete to the new style web pages. They were later replaced by the visual cues based approach, but this approach also used the DOM as an underlying technique. Recent approaches of web page segmentation perform segmentation of a web page by rendering the image of the web page, or creating a graph with the segments as the nodes of the graph but they do not focus on querying applications.Figure 1, consolidates the evolution between the web page generation technologies and segmentation methodologies. It represents evolution of a web-page segment from being a mere syntactical HTML fragment to a well-knit semantic region on the web page. As shown in the figure, the DOM-based techniques can be successfully applied to plain HTML pages.These solutions analyze the HTML DOM and extract information about the appearance of objects on the page and thereby, group HTML objects.These solutions fail with dynamic HTML pages. In case of dynamic HTML pages, the object hierarchy is often available, but it does not describe the layout and components semantically. When we apply the layout algorithm successively, we divide the page to smaller and smaller components, according to the natural visual hierarchy. For such pages, visual cues methods based on generic design heuristicsis a sought solution. The visual cues add to the capability of the algorithm to handle the evolving web page design.

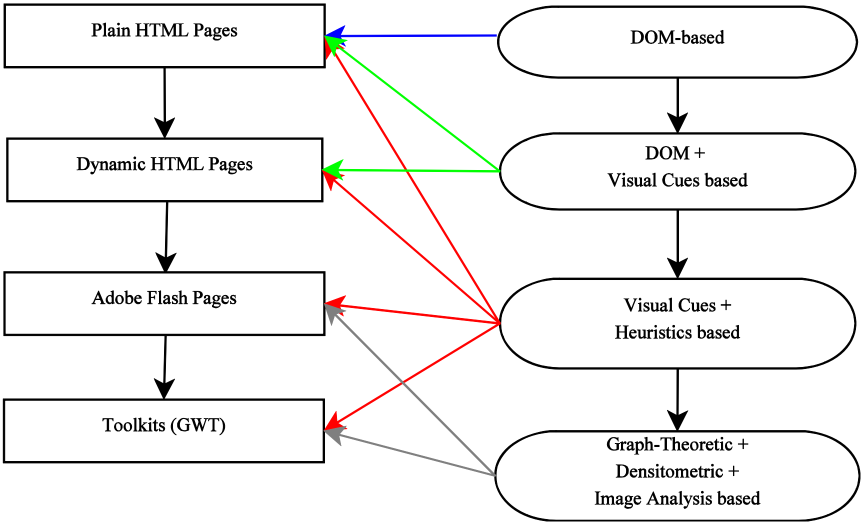


Fig. 1. Evolution of Web Technologies and Web Page Segmentation Methods

3 Web Segmentation Algorithms

3.1 Characteristics of a Good Web Page Segmentation Algorithm

A web page's structure and layout depend on different content type it represents or the taste of designer styling its content. Thereby main content position differs in variety of websites. Even there might be some content in page view that are besides each other but actually in DOM tree they are not in the same level or share same parent node. Finding the main content for querying in this situation where the content doesn't follow any specific rules for arranging and positioning elements needs complicated and expensive algorithms. We list the most desired characteristics in a webpage segmentation algorithm for improved querying:

- Algorithm should be able to simulate a user visiting the website [2].
- It should have high probability to find informative content because in most cases actual users in internet wish to query the informative areas and leave the non-informative segments [2].
- It should be capable to generate the hierarchical structure of a web page.
- The space and time complexity of the algorithm should be reasonable.

Figure 2, displays our target framework in the scenario of human-web interactions. It displays the characteristics desired in a good web-page segmentation algorithm and how they help in generating an improved query interface.

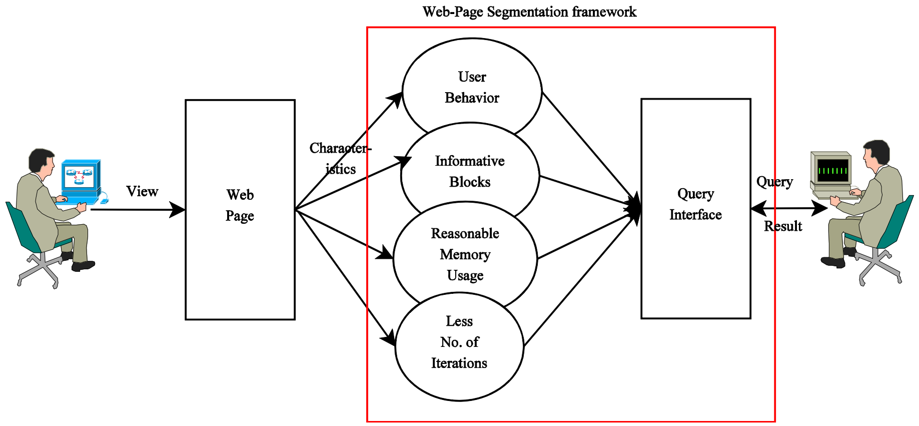


Fig. 2. The Web Page Segmentation and Query Interface Generation Framework

3.2 Classification Criteria

In this sub-section, we classify the various segmentation algorithms based on their underlying approach. The classification tree is captured in Figure 3. These algorithms can be broadly classified into three categories: the DOM-based algorithms, these algorithms are dependent on the rendering of the HTML elements based on the underlying or hidden DOM tree of the web page;methods based on visual cues, these can be further classified into methods keeping the DOM tree in vision and those combining it with heuristic design rules and the modern methodswhich include performing edge analysis over the image of the web-page, constructing a weighted graph that is segmented. Another performs shrinking and dividing operations on the web page. In the following sub-sections we discuss each one of these approaches, highlight the *VisHue* algorithm for web page segmentation and compare it with its counterpart VIPS [1] algorithm. For each of them we discuss the basic approach, existing works utilizing the method, strengths and weaknesses. The comparison is summarized in Table 2.

DOM-Based Algorithms. In general, similar to discourse passages, the blocks produced by DOM-based methods tend to partition pages based on their pre-defined syntactic structure, i.e., the HTML tags. Some simple experiments were performed in [21], where sub-trees tagged with <TITLE>, <P>, <H1>~<H3> and <META>were treated as blocks, but the results were not encouraging.

In some cases this approach can deal with "badly" presented pages. Since almost all blocks share the same length, there are no priorities for short blocks. As windows are overlapped, more blocks are likely to be extracted from a long document than VIPS [1]. However, a lot of web pages do not obey the W3C HTML specifications, which might cause mistakes in DOM tree structure. Moreover, DOM tree was initially introduced for presentation in the browser rather than description of the semantic structure of the web page. Hence, two nodes which may appear to be semantically related actually may not be related. Much recent work [11], [13], [14], and [17] try to extract the structure information from HTML DOM tree [1]. The segmentation by the DOM-based techniques is too detailed [4]. After partitioning, although each block represents some information, it usually does not provide complete information about a single semantic, and thus does not contain good expansion terms [4].

The weaknesses of this approach are:

- DOM is a linear structure, so visually adjacent blocks may be far from each other in the structure and divided wrongly.
- Tags such as <TABLE> and <P> are used not only for content presentation but also for layout structuring. It is difficult to obtain the appropriate segmentation granularity.
- In many cases DOM prefers presentation to content and therefore it is not accurate enough to discriminate different semantic blocks in a web page.
- The number of possible DOM layout patterns is virtually infinite, which inescapably leads to errors when moving from training data to Web-scale [20].

Visual Cues Based Methods. In the sense of human perception, it is always the case that people view a web page as different semantic objects rather than a single object. Actually, when a web page is presented to the user, the spatial and visual cues can help the user to unconsciously divide the web page into several semantic parts. Therefore, it might be possible to automatically segment the web pages by using the spatial and visual cues. Visual cues are very helpful to detect the semantic regions in web pages. Due to the 2-D logical structure, web pages could be partitioned in a 2-D style. A block is assumed to have a rectangle shape and is a closely packed region in the original page.

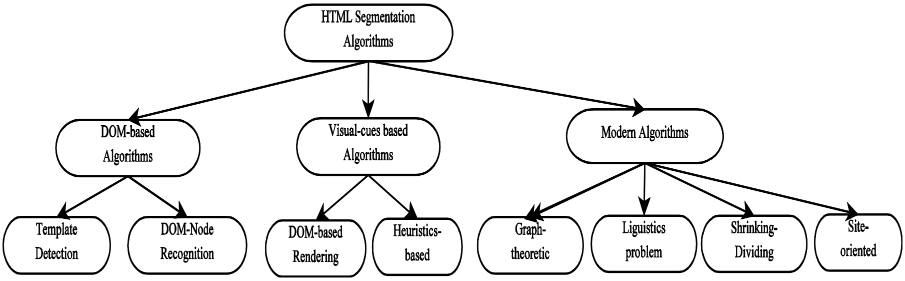


Fig. 3. Classification of Web-Page Segmentation Algorithms

DOM and Visual Cues Based Algorithm (VIPS) [1]. The VIPS (Vision-based Page Segmentation) algorithm extracts the semantic structure of a web page. This semantic structure is a hierarchical structure in which each node corresponds to a block. Each node is assigned a value (Degree of Coherence) to indicate how coherent the content in the block is. The VIPS algorithm makes full use of page layout features: first it extracts all the suitable blocks from the html DOM tree, and then tries to find the separators between the extracted blocks. Here, separators denote the horizontal or vertical lines in a web page that visually cross with no blocks. Finally, based on these separators, the semantic structure for the web page is constructed.

It tries to fill the gap between DOM structure and the conceptual structure of the webpage. The algorithm uses the content structure and tries to simulate how actual user finds a main content based on structural and visual delimiters. The DOM structure and visual information are used iteratively for visual block extraction, separator detection and content structure construction. Finally a vision-based content structure is extracted. In the VIPS method, a visual block is actually an aggregation of some DOM nodes. Unlike DOM-based page segmentation, a visual block can contain DOM nodes from different branches in the DOM structure with different granularities [4].

The blocks obtained from VIPS still have the varying length problem and suffer from lack of normalization factor. More importantly, it remains unclear whether the method would work on passage retrieval and no comparison is provided between this method and traditional passage retrieval methods such as windows, which can be naturally applied to web documents. A web page will first be passed to VIPS for segmentation, and then to a normalization procedure [13]. The blocking result is satisfactory but the algorithm does many loops to reach its desire granularity [2]. We also notice that, for those "badly" presented web pages, VIPS usually fails to partition them into semantic blocks and thus expansion terms are likely to be irrelevant. Also, some relevant long blocks produced by VIPS are ranked low since similarity measure tends to favor short documents.

VisHue Algorithm. The heuristics utilize geometry-related features present on aweb-page, and apply the rules in a greedy fashion to produce the segments. If the heuristics are carefully defined and are generic in nature they prove a strong methodology overcoming weaknesses like the solutions based on heuristics tend to local minima, or they involve a lot of trial and error effort when combined. When combined with the visual cues they give a generic approach for web page segmentation and generate a hierarchical structure for the web pages. Defining heuristics on the base of the web pages can prove more useful rather than the dynamic elements. For e.g.: A heuristic rule stating "headings at same level have same orientation" is more generic and applicable than stating "headings of blue color should be aligned left". Therefore, the former rule has a broader scope of application.

For *VisHue*, we focus on the following key points:

- A method independent of any underlying source code, web standard or web page generation language.
- Web pages may or may not have clear gaps distinguishing the content groups.
- Segmentation need not be too detailed and should be focused on developing an improved query interface.
- The segments that are created should be non-overlapping and capable of constructing a hierarchical structure.

We also assume that (i) Most of the webpages within a website have similar structure and (ii) The geometric patterns can be rendered to derive the inter-content relationship.

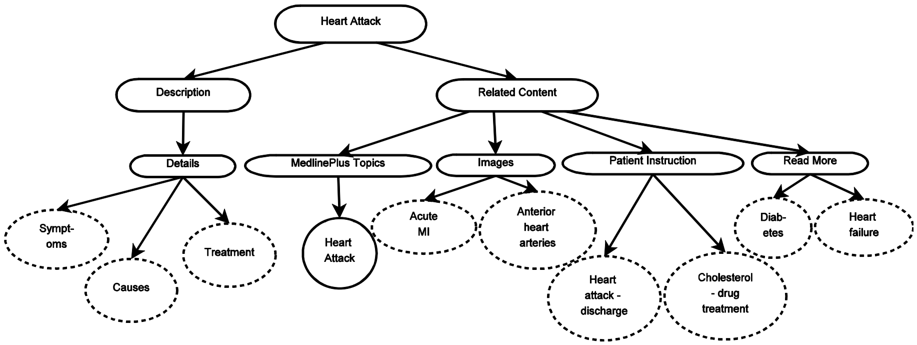


Fig. 4. The Resultant Content Structure

The *VisHue* algorithm creates a hierarchical structure of the web page which is semantically labeled and is suitable for improved querying. The relationships among the content nodes is evident which are perhaps most important for developing a good query interface. The approach constructs two structures; one is a skeleton of the web page called the *domain specific tree* where the domain refers to a website or a group of similar websites, for instance, "MedlinePlus medical encyclopedia". It defines a set of candidate labels for all the possible contents in the domain, renders the tree semantically and assigns labels to the nodes of the tree based on the candidate set to generate a labeled tree termed as the *tree of semantics*.This approach converts the extraction problem to an integration problem. By integrating the two trees we can reproduce the hierarchical structure with subheadings and headings of the page which are the labels of nodes in the tree.Figure 4, shows thehierarchical structure for the MedlinePlus encyclopedia page for the topic "Heart Attack". The dashed lines indicate that the node has more siblings but are not represented due to space constraint. This structure can be mapped to the schema of the web pages directly. We also take into account any differences that might occur amongst the various web pages like missing subheading etc. In the next section we explain how this structure helps in developing a better query interface.

Modern Methods. In this subsection we take up the modern algorithms for web page segmentation.

Graph theoretic segmentation [6]: This approach formulates the segmentation problem in a combinatorial optimization framework. It casts it as a minimization problem on a suitably defined weighted graph, whose nodes are the DOM tree nodes and the edges are the weights expressing the cost of placing the end points in same or different segments. It takes this abstract formulation and produce two concrete instantiations, one based on correlation clustering and another based on energy-minimizing cuts in graphs. Both these problems have good approximation algorithms.

The quality of segmentation in this algorithm depends heavily on the edge weights. The empirical analysis in the paper shows that the energy minimizing formulation performs better than the correlation clustering formulation. It proves that learning edge-weights from labeled data also produce appreciable improvements in accuracy. The segmentation algorithm is applied as a pre-processing step to the duplicate webpage detection problem.

The shortcoming of this approach is the identification of the DOM elements to create the graph. The graph construction is a tedious task, involving a lot of terminology and in-depth understanding. Hence, using such an approach for the purpose of enhancing the querying of a web page will be a complex task where such a high degree of precision in partitioning is not a priority. Moreover, unified query interfaces cannot be constructed as the graph will become more complex when many query interfaces will be considered together.

Site –Oriented Segmentation [7]: Since many pages belonging to a same web site share a common structure, look and feel, this approach hypothesizes that one can achieve a more accurate segmentation by taking all pages of the same web site into account.Based on this idea, the authors of [7] propose and evaluate a segmentation method which segments pages according to properties of the whole web site, instead of just information from a single page. The method adopts a DOM tree alignment strategy proposed for template detection [31, 33]. This method was developed especially for the so called *data-intensive w*eb sites, such as digital libraries, web forums, news web sites, etc. whose main focus is providing access to a large quantity of data and services [9]. These sites usually contain a large set of web pages which can be organized in a few tens of groups according to the regularity of their structure. This approach focuses on input to web search systems and other similar applications but depends on the DOM tree of each of the different web pages in the website.

Densitomeric Segmentation [20]: This approach builds methods from Quantitative Linguistics and strategies are borrowed from the area of Computer Vision.It utilizes the notion of text-density as a measure to identify the individual text segments of a web page, reducing the problem to a 1D-partitioning task. The distribution of segment level text density follows a negative hyper geometric distribution, described by Frumkina's Law. Their extensive evaluation confirms the validity and quality of the approach and its applicability to the Web. They define an abstract block-level page segmentation model which focuses on the low-level properties of text instead of DOM-structural information. The number of tokens in a text fragment (or more precisely, its token density) is a valuable feature for segmentation decisions.

The strengths of this approach lies in the fact that it reduces the page segmentation problem to a 1D-partitioning problem. It proposes a *Block Fusion algorithm* for identifying segments using the text density metric. It presents an empirical analysis of the algorithm and the block structure of web pages and evaluates the results, comparing with existing approaches. It shows the application of the methodology to the field of near-duplicate detection.

Image analyses of the web page [8, 10]: In this approach a layout for segmentation of the web page is generated by performing edge analysis on the GUI image (or its transformation).It assumes that the main objects are outlined so that there is a border between them. It seeks for areas containing information, and groups them into distinct layout elements. This technique gives a high level layout; thereby segmenting the page to its main components.This approach uses only the visual information and does not apply any semantic analysis to group or ungroup elements. It finds these layouts recursively deeper into the page. The recursive process continues until down to the level of individual elements. Deeper in the hierarchy, this task becomes more difficult because the objects we separate become smaller. Thus, the edges are denser and tend to merge.

After segmentation it uses text detection and applies OCR, which also gives information about the meaning of a layout object. The text information is important for later semantic analysis of the page content. Hence, though the approach segments the web page visually, it does not pay attention to the semantic grouping. Such a technique will fail in case of web-pages with semantically related yet scattered content.

The visual cues and heuristics based method*VisHue*, independent of any standard or model over the web andare applicable to plain HTML pages, dynamic HTML pages, flash pages, or those generated by any of the web toolkits, since it does not depend on any source codes. Since it iterates only till each part of the webpage is covered by some node of the tree, the time complexity of the algorithm is reasonable.The tree stored in the memory comprises of just the headings and subheadings and the height of the tree is directly proportional to the levels of nesting of the subheadings within the page. This number is bounded. The number of leaf nodes is bounded by the distinct blocks in the web page. Hence, the space complexity of the algorithm is reasonable. Moreover, its capability to construct a hierarchical structure makes itappropriate for query interface design. We highlight the strengths of the *VisHue* algorithm in Table 1.

Table 1. Comparison between VIPS and VisHue

Characteristics	VIPS	VisHue algorithm
Underlying technique	Recognition of Visual Cues using basic DOM elements	Visual cues and heuristic rules
Precision	Basic DOM elements	Visible segments on the web page
Application	Block based web search	Advance querying
Authenticity	Non-overlapping blocks.	Better blocks formulation than VIPS
Data structure	Hierarchical tree of all the blocks	Hierarchical tree of headings or labels in the web page
Space complexity	Entire structure needs to be stored	Only the headings of the blocks are stored
Time complexity	Recursive partitioning till the basic DOM elements are found	Less, no attempt to reach the indivisible DOM elements.
Language or standard dependency	DOM dependent	None

Table 2. Comparison Summary Between Various Web Page Segmentation Approaches

	Template Detection	**DOM-node Recognition**	**Graph-theoretic**	**Image Analysis**	**Linguistic approach**	**Shrinking and dividing**	**Site-oriented**
Underlying technique	DOM elements	DOM elements	Weighted graph	Edge analysis on web page image	Token density of a text fragment	Image processing and web page characteristics	DOM tree alignment
Precision	Basic DOM elements	Basic DOM elements	Aggregates at the level of internal nodes	Individual elements	Blocks based on text density	Indivisible sub-images	Basic DOM elements
Application	Link structure analysis	Link structure analysis	Duplicate Detection	Not specified	Near duplicate detection	Phishing page detection	Segment aware web search
Authenticity	Syntactic segments	Syntactic segments	Energy-minimizing cuts	Not specified	Not specified	Not specified	Not specified
Data structure	DOM-tree	DOM-tree	Regions of visual content	GUI image	Statistics: block fusion algorithm	Sub-images of content blocks	Tree combining all the DOM trees
Space complexity	DOM elements are stored	DOM elements are stored	Not specified	Not specified	Minimal	Not specified	Not-specified
Time complexity	Recursive partitioning	Recursive partitioning	Not specified	Recursive partitioning	15ms per page	Serves real time	Not specified
Language or standard dependency	DOM dependent	DOM dependent	Depends on DOM tree	None	None	None	DOM-dependent

VisHue algorithm scores over its counterpart algorithm VIPS on the following features:

- It addresses the visual design heuristics within a web page, whereas the VIPS algorithm gives heuristic rules which are DOM based. For a web page not based on the DOM elements, latter will fail.
- Some of the heuristics of VIPS like,if all the child nodes of a node are text nodes then the node should not be further segmented; implies that a web page where the contents are organized under a single node like MedlinePlus medical encyclopedia this approach will fail.
- *VisHue* labels the hierarchical nodes semantically whereas the VIPS do not assigns labels to the block-hierarchy it constructs.
- The block-based search based on VIPS does not confine the search space for a user query though the blocks are returned as results whereas the labels of the *VisHue* algorithm reduce the search space for user queries significantly.

4 Improved Query Interface Using the VisHue Algorithm

4.1 Web Page Segmentation and Improved Querying

Currently, information on the web may be discovered primarily by two mechanisms: browsers and search engines. Existing search engines such as Yahoo, Google service millions of queries a day. Yet it has become clear that they are less than ideal for retrieving an ever-growing body of information on the Web. This mechanism offers limited capabilities for retrieving the information of interest; still burying the user under a heap of irrelevant information [26]. Also the documents on the web are not well-structured so that a program can reliably extract the routine information that a human wishes to discover. These searches are generic. For example, if a user wishes to find an article authored by a person X. The query results will show all articles with an occurrence of X. Such results are not relevant for the end-users as they do not expect such a generic set of results. Hence, we conclude that there is a need for an in-depth querying of the web pages. And provide users with results that are from specific segments of the web page.

There have been works like the *Block-based Search* [4], where the webpage is segmented into semantic blocks and the importance values of the blocks are labeled using a block importance model [2]. Then the semantic blocks, with their importance values, are used to build block-based Web search engines [1], [3].But these blocks do not improve the query interfaces. In *Object-Level Vertical Search*, all the web information about a real world object or entity is extracted and integrated to generate a pseudo page for this object. These object pseudo pages are indexed to answer user queries, and users can get integrated information about a real-world object in one stop. This object-level vertical search technology has been used to build Microsoft Academic Search (http://libra.msra.cn) and Windows Live Product Search (http://products.live.com).Another type of search called *Entity Relationship Searc*hdeploys an Entity Relationship Search Engine in the China search market called Renlifang (http://renlifang.msra.cn). In Renlifang, users can query the system about people, locations, and organizations and explore their relationships. These entities and their relationships are automatically mined from the text content on the Web. If the query terms scatter at various regions with different topics, it could cause low retrieval precision. It can be argued that a web page with a region of high density of matched terms is likely to be more relevant than a web page with matched terms distributed across the entire page even if it has higher overall similarity.

Keeping the above discussion in mind, and observing the lack of a well formed query interfaces for the encyclopedia like websites, prompted us to utilize the work of segmenting the web page using the *VisHue* algorithm for developing a query interface. Web page segmentation empowers the user to expand his querying horizons by providing him tags or labels of the subtopics within the page that can be queried individually. For instance, a disease name in the user's search criteria will have a web page containing details about it, alongside; it can be a part of the symptoms, causes of another disease or a diagnosis of some test. When an end-user queries the encyclopedia, he is presented with all the results along with the web page about it. All these results may or may not be of relevance to him. Instead, providing him a specific set of results, such as if the query is "heart attack as a symptom", will be more beneficial. Hence, the query is reformulated in terms of occurrence of the queried term within the encyclopedia. Our proposed query interface combines both the object-based as well as the entity-based search. It exploits the relationship between different content segments in a web page and can query specific regions of the web page.

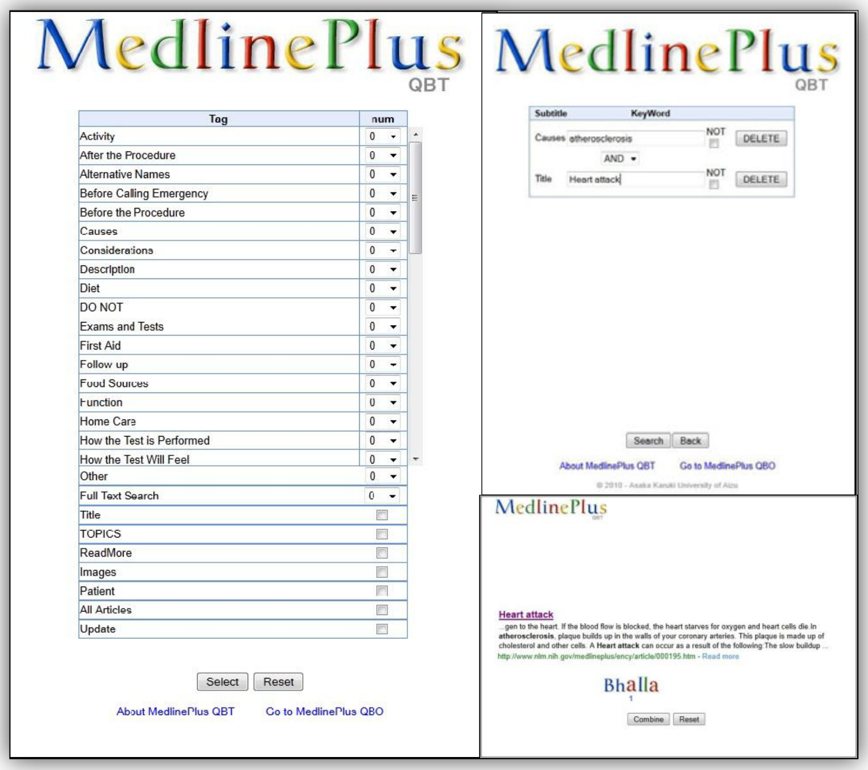
4.2 Query by Segment

Query by Segment is a query interface for the medical encyclopedia by MedlinePlus [3]. We refer Query by Segment as QBT (Query by Tag) in the study. It is an interface for formulating and retrieving query results by various subheadings and headings of different content segments in the web page belonging to the encyclopedia. It utilizes the content structure generated by the *VisHue* algorithm method mentioned in previous section and uses the node labels as query fields within which search can be performed. It allows the user to confine his search to specific regions within a page. It provides only the relevant results for a user query. We compare QBT with standard keyword search available at MedlinePlus [3]. Let us suppose that an end user wants to search "nausea" as a symptom. Using standard search MedlinePlus, displays search results where "nausea" has an occurrence in *side effect of medication* besides in *symptoms*. On the other hand, using QBT, we can just display *nausea* from *symptoms*. Further comparison is given in next subsection.

QBT and the Hierarchical Structure of a Web Page. The node labels of the hierarchical structure are mapped to query fields in the QBT interface. The user can select the sub-heading he wishes to query and also for a keyword can specify scope of search. Once the user selects the subheading or subtitle of a segment to query he or she can also choose to search the related topics, related content, read more etc. Next he or she moves to the second screen where she or he inputs the value of the fields (segment headings) selected in the previous screen. In the figure 5, we see that the user enters the values for "causes" and "title" selections. On this screen the user has the provision to delete an attribute, perform an "OR" or an "AND" operation on the attributes. Once, his or her query is formulated, he or she clicks on the "search" button and is presented with the results that are best fit for his query. In our example displayed in Figure 5, the results are displayed where “Heart Attack” is found in the title and “atherosclerosis” in the “causes” as desired by the user.

We map the design of the QBT from the hierarchical tree of the web page. The child nodes of the description and related content become the search areas within which one can search. Their children define the subspaces that can be searched. For e.g.: If a user wishes to search a symptom X for a disease say “Heart Attack” checks the title to be searched with the keyword “Heart attack” and some keyword for symptom. The content structure enables the interface with a smart search (as required by the above example) by incorporating the following points:

- The left siblings’ limit the options for the right siblings in the query interface. If any user selects one sub-heading, it highlights the other sub-headings that co-occur with it in the web pages within the encyclopedia.
- The child nodes of the main content (MC) and related content (RC) define the complete search space for a user and the leaf nodes define the fine lines of search for the user within them.
- The candidate set of labels define an exhaustive list of sub-headings that can define the searchable areas within the encyclopedia.



Tag	num
Activity	0
After the Procedure	0
Alternative Names	0
Before Calling Emergency	0
Before the Procedure	0
Causes	0
Considerations	0
Description	0
Diet	0
DO NOT	0
Exams and Tests	0
First Aid	0
Follow up	0
Food Sources	0
Function	0
Home Care	0
How the Test is Performed	0
How the Test Will Feel	0
Other	0
Full Text Search	0
Title	□
TOPICS	□
ReadMore	□
Images	□
Patient	□
All Articles	□
Update	□

Subtitle	KeyWord		
Causes	atherosclerosis	NOT □	DELETE
	AND		
Title	Heart attack	NOT □	DELETE

Heart attack
...gen to the heart. If the blood flow is blocked, the heart starves for oxygen and heart cells die.In **atherosclerosis**, plaque builds up in the walls of your coronary arteries. This plaque is made up of cholesterol and other cells. A **Heart attack** can occur as a result of the following:The slow buildup ...
http://www.nlm.nih.gov/medlineplus/ency/article/000195.htm - Read more

Fig. 5. The MedlinePlus QBT Interface

A Tour of the QBT Interface. We take a brief look at the QBT interface. The interface is composed of three screens, the subtitle selection screen; screen to enter keywords for query and finally the results screen. The initial screen displays a table virtually divided into two parts one describing the various headings or sub-topics of different segments within the encyclopedia page and the other part defines the scope of search in which the user wishes to search a keyword shown in Figure 5 left side. For instance, here the user selects "1" occurrence of "causes" it highlights all the co-occurring segments, the user selects "title" space to search another keyword and clicks "select", he or she is presented with the query formulation screen, where the user enters the values for his or her selections; the user is presented with an exhaustive list of values he or she may enter. The fields have a hint based auto-completion facility. The second and the final screens are shown in Figure 5 right side.

Table 3. Comparison of Querying Features between QBT and Traditional Keyword Search

S.No.	Features	QBT	Keyword Search
1	**Direct Answers**	Precise, direct answers returned	A set of articles with an occurrence of keyword(s) is returned
2	**Query Capability**	Focused querying	Limited
3	**Retrieval Units**	Text snippets along with article URLs	Article URLs
4	**Aggregate Queries**	Various querying operations like AND, OR and NOT	Not Possible
5	**Usefulness**	Exact and relevant results	Large number of results are presented which need to be sorted by user
6	**Easy to use Interface**	Labels are self-explanatory and there is provision of defining the scope of search for a given keyword	Simple and Advanced option for entering the keyword(s)

4.3 Performance of the QBT Interface

In this sub-section we evaluate the performance of the QBT interface, w.r.t the traditional keyword search available for the MedlinePlus medical encyclopedia. We draw a qualitative and quantitative analysis of the performance of the interface and exhibit its efficiency over the existing method to query the encyclopedia. We also differentiate the query formulation of the two methods.

Implementation Details. The QBT interface is implemented on Windows 7, 64-bit OS. Apache HTTP Server is used as a platform to run the application, PHP scripting language is used for user interface (UI) design and IBM DB2 database is used. Table3 presents a qualitative analysis of the QBT interface w.r.t the keyword search. It lists the features of a useful query interface for an end-user to perform efficient information retrieval.

Table 4. Query Formulation in QBT and Traditional Keyword Search

S. No.	User Query	QBT Query	Keyword Search
1	Cases where patient has hypertension but not high blood pressure	Symptom: "Hypertension" Symptom: NOT "High blood pressure"	Search has no provision of negating one of the keywords
2	Cases for patient to stop certain activities before a test (can resume normal activities after it)	Before Procedure: "Stop" After Procedure: "Normal"	Keyword search with "stop" and "normal" keywords
3	Heart attack caused by high blood pressure	Cause: "High blood pressure" Symptom: "heart attack"	Search for keyword "heart attack" and "high blood pressure"
4	Poisoning caused by eating fish	Food Source: "Fish" Side Effect: "Poisoning"	Search all articles for keywords "fish" and "poisoning"

Table 3 shows the QBT interface is a far better approach to query or search the medical encyclopedias. It has the capability to support aggregated queries and complex queries where user can find articles with occurrence of a keyword and negation of another keyword. It is an easy to understand interface that provides the end-user precise answers for his queries. All these features make it a much powerful interface for medical domain.

Table 4 shows how a user query is understood by either of the interfaces. The QBT interface understands the user query in perhaps the most natural way. Whereas the keyword search just picks up the keywords and fails in case of aggregate or negation like queries. We perform a quantitative analysis of the performance of the QBT interface with a small set of queries. We observe the performance of both querying mechanisms on these queries and present the analysis in Table 5. The existing keyword search in MedlinePlus is generic to the entire website. Hence, we enter a keyword and later sort the results belonging to the medical encyclopedia. Table 5 shows a comparison between the number of results and number of relevant results and calculates their relative percentage. We also compare the rank at which a resultdisplayed by QBT occurs in case of the keyword search. Web ranking is critical for information retrieval methods as stated in Section 2. The reduction in the search space is significant in case of the QBT interface whereas the user may not be able to confine his search space in case of simple keyword search. We calculate the percentage of content searched with respect to a total of 4000 web pages within the MedlinePlus medical encylopedia. Hence, we conclude that QBT narrows down the results and displays only the relevant results to the user and cuts down the processing time of these queries.

Table 5. Quantitative Performance Analysis of the QBT and Keyword Search for User Queries in Table 4

User Query	Parameters	QBT	Keyword Search
1	No. of Results	4	380
	No. of Relevant Results	4	4
	Relative Ranking	All 4	No result in top 10
	Relevant Results (%)	100	1.05
	Contents Searched (%)	0.01	100
2	No. of Results	15	523
	No. of Relevant Results	15	15
	Relative Ranking	All 15	No result in top 10
	Relevant Results (%)	100	2.87
	Contents Searched (%)	2.5	100
3	No. of Results	3	145
	No. of Relevant Results	3	3
	Relative Ranking	All 3	No result in top 10
	Relevant Results (%)	100	2.06
	Contents Searched (%)	2	100
4	No. of Results	3	85
	No. of Relevant Results	3	3
	Relative Ranking	All 3	No result in top 10
	Relevant Results (%)	100	3.53
	Contents Searched (%)	0.5	100

5 Discussion and Future Work

The majority of the existing methods for web-page segmentation compute structural similarity using features derived from HTML codeor DOM tree representation of web pages [1], [5], and [11]. Only little work has been done to compare web pages based on their visual structure [6].From a web user's perspective, however,the visual structure of a web page is more discriminating than the structure of its source code: The fundamental reason is that the process of rendering a web page is a non-injective, and hence lossy mapping is done from a one-dimensional code fragment into a two-dimensional arrangement, where the same visual appearance can be generated by very distinct HTML code fragments. With ever more complex web pages and more available HTML options to create the same design, structural similarity as perceived by web users can be only reliably determined from a web page's visual rendering [18].

The paper summarizes all the existing works in the field and highlights a detailed comparison between them. It gives an account of the on-going work on formulating a generic heuristic design-ruleandvisual cues based *VisHue* algorithm for web page segmentation which is more beneficial than the existing methods for querying purposes and constructs a corresponding hierarchical structure. It discusses the QBT interface which is designed by using this approach. Future work will include improving the QBT interface; making it more generic, dynamic and intelligent in nature. A practical evaluation of the proposed *VisHue* algorithm and a comparison of performance between the QBT and the block based search in terms of reliability and scalability.

6 Summary and Conclusions

The present study considers a detailed account of the existing approaches for web-page segmentation. It gives a comparison among these approaches. Since, the face of the web is continuously changing the need for such approaches are ever growing. By understanding the current techniques, the scope for improvements can be clearly understood. Combining heuristics with the visual rendering of the web page for web page segmentation can prove to be a turning point in the need for language independent solutions for web-page segmentation and for improving existing query interfaces. The medical domain has a need to evolve in terms of making the available information accessible to the end-users in a user-friendly manner. The need for advanced query interfaces that provide an in-depth querying persists. Query by Segment or Tag (QBT) is an attempt in this direction. It aims at returning the users the desired results from the designated parts of the web-page rather than complete web page results.

References

1. Cai, D., Yu, S., Wen, J., Ma, W.-Y.: Extracting Content Structure for Web Pages based on Visual Representation. In: Zhou, X., Zhang, Y., Orlowska, M.E. (eds.) APWeb 2003. LNCS, vol. 2642, pp. 406–417. Springer, Heidelberg (2003)
2. Asfia, M., Pedram, M.M., MasoudRahmani, A.: Main Content Extraction from Detailed Web Pages. International Journal of Computer Applications (IJCA) 11 (2010)
3. `http://www.nlm.nih.gov/medlineplus/encyclopedia.html`

4. Cai, D., He, X., Wen, J.-R., Ma, W.-Y.: Block-based Web Search. In: Proc. of SIGIR (2004)
5. El-Shayeb, M.A., El-Beltagy, S.R., Rafea, A.: Extracting the Latent Hierarchical Structure of Web Documents. In: SITIS (2006)
6. Chakrabarti, D., Kumar, R., Punera, K.: Graph-Theoretic Approach to Webpage Segmentation. In: WWW 2008 / Refereed Track: Search - Corpus Characterization & Search Performance, Beijing, China (2008)
7. Fernandes, D., de Moura, E.S., da Silva, A.S.: A Site Oriented Method for Segmenting Web Pages. In: SIGIR 2011, July 24-28 (2011)
8. Cao, J., Mao, B., Luo, J.: A segmentation method for web page analysis using shrinking and dividing. International Journal of Parallel, Emergent and Distributed Systems 25(2), 93–104 (2010)
9. Bohunsky, P.: Visual Structure-based Web Page Clustering and Retrieval. In: WWW 2010, Raleigh, North Carolina, USA, April 26-30 (2010) (Poster)
10. Pnueli, A., Bergman, R., Schein, S., Barkol, O.: Web Page Layout via Visual Segmentation. HP Laboratories
11. Chakrabarti, D., Kumar, R., Punera, K.: Page-level template detection via isotonic smoothing. In: 16th WWW, pp. 61–70 (2007)
12. Kao, H.-Y., Ho, J.-M., Chen, M.-S.: WISDOM: Web intrapage informative structure mining based on document object model. TKDE 17(5), 614–627 (2005)
13. Bohunsky, P.: Visual Structure-based Web Page Clustering and Retrieval. In: WWW 2010, Raleigh, North Carolina, USA, April 26-30 (2010) (Poster)
14. Pnueli, A., Bergman, R., Schein, S., Barkol, O.: Web Page Layout via Visual Segmentation. HP Laboratories
15. Chakrabarti, S., Joshi, M., Tawde, V.: Enhanced topic distillation using text, markup tags, and hyperlinks. In: 24th SIGIR, pp. 208–216 (2001)
16. Kao, H.-Y., Ho, J.-M., Chen, M.-S.: WISDOM: Web intrapage informative structure mining based on document object model. TKDE 17(5), 614–627 (2005)
17. http://adam.about.net/encyclopedia/
18. http://www.drugs.com/medical_encyclopedia.html
19. http://www.mgo.md/encyclopedia.cfm
20. http://www.umm.edu/ency/
21. Hong, J., He, Z., Bell, D.A.: An evidential approach to query interface matching on the deep Web. Information Systems Journal 35(2) (2010)
22. Kohlschütter, C., Nejdl, W.: A Densitometric Approach to Web Page Segmentation. In: CIKM 2008, October 26-30 (2008)
23. Crivellari, F., Melucci, M.: Web Document Retrieval Using Passage Retrieval, Connectivity Information, and Automatic Link Weighting. In: TREC-9 Report, In The Ninth Text Retrieval Conference (TREC 9), (2000)
24. Nie, Z., Wen, J.-R., Ma, W.-Y.: Webpage Understanding: Beyond Page-Level Search. Sigmod Record 37(4) (2008)