COMPUTER
ENGINEERING
PROGRAM

Middle East Technical
University
Northern Cyprus
Campus

eMINE Technical Report Deliverable 0 (D0),

March 2011

Web Page Segmentation: A Review

Yeliz Yesilada
Middle East Technical University,
Northern Cyprus Campus,
Kalkanlı, Güzelyurt, TRNC,
Mersin 10, TURKEY

Web pages are typically designed for visual interaction. In order to support visual interaction they are designed to include a number of visual segments. These visual segments typically include different kinds of information, for example they are used to segment a web page into a number of logical sections such as header, footer, menu, etc. They are also used to differentiate the presentation of different kinds of information. For example, on the news site they are used to differentiate different news items. This technical report aims to review what has been done in the literature to automatically identify such segments in a web page. This technical report reviews the state of the art segmentation algorithms. It reviews the literature with a systematic framework which aims to summarize the five Ws – the 'Who, What, Where, When, and Why' questions that need to be addressed to understand roles of web page segmentation.

METU
NCC

eMINE

The World Wide Web (web) has moved from the Desktop and now is ubiquitous. It can be accessed by a small device while the user is mobile or it can be accessed in audio if the user cannot see the content, for instance visually disabled users who use screen readers. However, since web pages are mainly designed for visual interaction; it is almost impossible to access them in alternative forms. Our overarching goal is to improve the user experience in such constrained environments by using a novel application of eye tracking technology. In brief, by relating scanpaths to the underlying source code of web pages, we aim to transcode web pages such that they are easier to access in constrained environments.

Acknowledgements

The project is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with the grant number 109E251. As such the authors would like to thank to (TÜBİTAK) for their continued support.

Contents

Middle East Technical University,
Northern Cyprus Campus,
Kalkanlı, Güzelyurt, TRNC,
Mersin 10, TURKEY

Tel: +90 (392) 661 2000
http://www.ncc.metu.edu.tr/

Corresponding author:
Yeliz Yesilada
Tel: +90 (392) 661 2994
yyeliz@metu.edu.tr
http://metu.edu.tr/~yyeliz

1 Introduction

Web pages are typically designed for visual interaction. In order to support visual interaction they are designed to include a number of visual segments. These visual segments typically include different kinds of information, for example they are used to segment a web page into a number of logical sections such as header, footer, menu, etc. They are also used to differentiate the presentation of different kinds of information. For example, on the news site they are used to differentiate different news items. However, unfortunately the underlying source code is not encoded in such a way to differentiate the visual segments. They are typically encoded for visual consumption and not for machine processing. For example, Figure 1 shows a screenshot of a news site. If we look at the visual rendering of the page, the segments are clearly differentiated. For example, on can easily look at the visual rendering and can differentiate the header which has red background colour, top menu which has black and grey background colour, headline news which has a bigger image and a larger text for the header, etc. However, when we look at the source code we cannot see such kind of clear segmentation or pattern in the source code. We can find list items, paragraphs, etc. However, identifying such segments can be very useful for different fields, for example, web pages can be properly displayed or repurposed for mobile devices, blind users can easily access them with their screen readers, search engine can use such implicit information to provide a better search result, etc.

If we consider the design and delivery process of web pages, segmentation can be done at different phase of web page development. There are some work that aims to annotate web pages or create web pages that already includes information about segments. For example, [22] proposes a technique that composes web pages as a number of segments. Alternative to this approach is to segment web pages after they are designed and developed. This could be done in two ways, manually or automatically. There are a number of work that aims to segment web pages manually, for example [6, 63] proposes a tool that can be used to manually annotate segments in a web page. Even though this could generate good quality annotations, there are a number of problems:

There are a number of work that aims to identify the segments in a web page manually, however in this work we are interested in automatic segmentation of web pages. There are also some issues:

1. it is labor-intensive;
2. it is error-prone;
3. does not scale, especially with dynamic content, the manual segmentation needs to be updated very often;

Alternative to manual evaluation is automated evaluation. There are a number of work in the literature that aims to automate the process of segmentation, but the segmentation of web pages is not a straightforward task and there are a number of challenges associated with it [33]. These challenges include:.

1. the ambiguity in defining the boundary of a segment – the ambiguity in terms of source code and the visual presentation;
2. the heterogeneity of web pages – different web sites have different layout;
3. the parameters/modifications needed to train an algorithm;
4. the performance, the time, speed it would take to process a web page;
5. issues with dynamic web pages, Javascript, dynamic updates, web 2.0;

6. copyright issues as the web page is actually changed by a third party;
7. web pages are not properly marked up, even TIDY cannot tidy up the page;
8. dynamic content adds extra challenge – pages include different parts with varying freshness and sharability requirements, fragments are updated with different rate and frequency.

This technical report aims to review what has been done in the literature to automatically identify such segments in a web page. This technical report reviews the state of the art segmentation algorithms. It reviews the literature with a systematic framework which aims to summarize the five Ws – the 'Who, What, Where, When, and Why' questions that need to be addressed to understand roles of web page segmentation. Our literature review shows the following:

Why did it happen? our literature review shows that the segmentation of web pages has been proposed in different fields for different reasons. In summary, the fields where segmentation is proposed is as follows: mobile web [78, 70, 75, 54, 46, 72, 73, 42, 68, 39], voice web [56], web page phishing [1, 20, 67, 50], duplicate detection [43, 21], information retrieval [18, 17, 15, 19, 80, 48, 8], image retrieval [40, 14, 65, 16, 33], information extraction [18, 17, 48, 11, 12, 13, 64], user interest detection/ sensemaking tasks [51, 27], visual quality evaluation [69], web page clustering [44, 77], caching [58, 59], archiving [60], publishing Dynamic Content [22], semantic annotation [55] and web accessibility [53, 9, 52].

Who is it about? segmentation is proposed to benefit different web users which include mobile web users, disabled users, search engine users.

How did it happen? different algorithms are proposed which use different techniques – graph-based algorithms [79], ranking algorithms, especially based on PageRank [78], heuristics-based algorithms [3, 54, 11, 12, 13], rule-based algorithms [18, 17, **?**, 19, 80], text-based algorithms [43], image processing [20], psychology based algorithms [76, 71],

machine learning algorithms [53, 9, 52, 7], clustering-based algorithms [4] and pattern matching algorithms [42, 68] and finally there are some custom algorithms [37, 35, 39, 27, 34, 24, 51, 70, 64, 77, 8].
Where did it happen? when we look at the web delivery method, web pages can be augmented in three different places: on the server side, on the client or on a proxy. When we look at the literature, we can find work that has been done on all these three different locations. There are some work that does the segmentation on the server side [2, 73, 26, 20, 15, 22, 40, 14, 65, 16] some on the proxy [74, 46, 37, 72, 73, 74, 26, 25, 42, 68, 76, 71, 7, 39, 6, 63] and some on the client-side[54, 79, 78, 35, 3, 56, 26, 21, 64, 55, 3, 18, 17, 61, 53, 9, 52].
When did it happen? The oldest work we found about segmentation is 2000 and it was first proposed for web accessibility – improving browsing experience of blind screen reader users [6, 63].
What happened? Finally, when we look at the evaluation of the proposed systems and algorithms, different metrics are used which include the following: Precision and recall [79, 46, 76, 71, 39, 65, 70, 11, 12, 13, 64, 64, 48], success rate and accuracy [3, 73, 26, 25, 42, 68, 76, 71, 21, 18, 17, 18, 17, 51, 44], simulation [37], user evaluation - Simulation [39, 75, 51, 27], task-based user evaluation [72, 27, 6, 63], execution time or speed or output size [72, 73, 74, 73, 56, 26, 25, 20, 76, 71, 39, 60] , comparison of algorithms [20, 76, 71, 18, 17, 40, 43, 21, 4], small set testing [7, 18, 17], specific task experiments, for example information retrieval [18, 17, 15, 19, 80, 8] or data mining experiments [77].

In the rest of this technical report, we discuss each of these questions in detail and the rest of the report is structured as follows: Section2 summarises and discusses the terms used in the literature to refer to a segment in a web page. Section 3 summarises the work that has been done to identify the role of the segments and their importance.

2 Segmentation Vocabulary: Terms Used in the Literature

Different published work use different terms to refer to a segment in a web page. Even though different terms are used they all refer to logical segments in a web page. Here we systematically look at the vocabulary used in the literature. The following list shows the vocabulary used in the literature.

Web elements [79, 78] refers to components of web pages as Web elements. A web page is made up of hundreds of basic elements. For example, the authors indicate that the functional role of each element is different. They use image as an example, an image can be a banner, advertisement or a picture of a news article.

Logical unit/section [54] refers to sections of web pages as logical units/sections of web pages. [35] refers to sections of web pages as semantically related group of elements. [33] uses the term section to refer to segments of web pages.

Block [46, 72, 73, 74, 26, 25, 20, 61, 75, 40, 14, 65, 16, 33, 67, 66, 50, 51, 43, 69, 4, 70, 60, 48, 77, 53, 9, 52] use the term block. [3] refers to them as information blocks which is defined as closely related content, each of which forms topic within the web page. They further classify the blocks as atomic (non-dividable block) and composite (can contain other composite or atomic blocks). [26, 25] refer to header, footer, sidebars

and body as the higher-level content blocks. [20] indicates that "a web page is composed of blocks, every visible element in HTML is displayed in its rectangular area. Labels, images, tables even flashes are all possessed of the basic parameters: width and length. So with a rectangle area, we can locate any visible element in web pages". [18, 17] define a block as the semantic part of the web page. [18, 17, 15, 19, 80] indicate that most people consider a web page as the smallest undividable unit but in fact it is inappropriate to represent a single semantic unit, therefore they propose a block to represent a single semantic unit. [4] uses the term block to refer to define a group of web contents that share a coherent topic, style and/or structure. They also indicate that they use the term *web content* to refer to the smallest indivisible content of a web page which is usually represented as a leaf node in the DOM tree.

Sub-page [37] views a web page as a collection of different small web pages structured together in a layout. These sub-pages can fit in the small screen that are meaningful units. [72] also refers to them as sub-pages as later than they use them for presenting purposes. A web page can be considered as a collection of sub-pages/blocks, etc [37].

Segments [56] indicates that when a user checks a page they can easily understand the segmentation and the structure of the page. [21] defines a segment as "is a fragment of HTML, which when rendered, produces a visually continuous and cohesive region on the browse window and a has a unified theme in its content and purpose". [39] refers to them as segments but they also refer to them as objects. [33] uses a number of terms and one of them is segment. [64] defines a segment as a cohesive region of a web page. [55] also refers to them as segments.

Component [42, 68] refers to the blocks of web pages as web page components. They define web page components as "basic units for transcoding which can be extracted through syntactic analysis of the page's HTML source code".

Coherent unit [76, 71] refers to blocks as coherent units.

Coherent region [7] refers to blocks as coherent region.

Objects [39] refers to them as segments but they are used interchangeably. [34] defines basic and composite objects. They define a basic object as "the smallest unit of a web page which cannot be further subdivided" and they define a composite object as " Authors group basic objects together to achieve a major function such group of objects is called composite objects, composite objects can then be grouped together onto a more complex one". [24] proposes a Function-Based Object model (FOM Model). According to this model, a web page consists of basic (smallest information body that cannot be further divided, only as a whole can perform certain functions) and composite objects (is a set of objects (both basic and composite) that perform certain functions together). [44] also refers to blocks as objects. They mainly focus on identifying common objects in web pages such as header, footer, left and right side bar and the main content.

Page unit/data unit/context unit [27] defines data units as web page fragments ranging from a single word to a complete page.

Fragment [58, 59] defines a fragment as "a fragment is a portion of a web page which has a distinct theme or functionality and is distinguishable from the other parts of the page". They further define *interesting fragments* as fragments fragments that has good sharability with other pages served from the same web site or it has distinct lifetime characteristics. [6, 63] also uses the term fragment to refer to the elements that are typically visually fragmented on a web page. [22] also refers to them as fragments. According to [22] web pages are constructed from simpler fragments and fragments may recursively embed other fragments. According to [22], they have a definition which is from the archiving perspective, and indicates that "fragments typically represents part of web pages which change together; when a change to underlying data occurs which affects several pages, the fragments affected by the change can easily be identified."

Area [11, 12, 13] refers to them blocks but they also use the term interesting areas on the page.

Pagelet [8] defines pagelets both semantically and syntactically. Their semantic definition is "a pagelet is a region of a web page that (1) has a single well-defined topic or functionality; and (2) is not nested within another region that has exactly the same topic or functionality" and their syntactic definition is "An HTML element in the parse tree of a page p is a pagelet if (1) none of its children contains at least k hyperlinks; and (2) none of its ancestor element is a pagelet".

3 The Function and Importance of the Segments

In the literature, some work also looks at understanding the role and the functions of the segments. For example, some segments in a web page has specific roles, for example some of them are used as headers, footers and some are used as the main content or headline in a web page. In this section, we review the literature and identify the work that has been done to understand the role of the segments.

- [79] proposes an algorithm based on random walks that classifies elements of Web pages into five categories which are Content (C), Related Links (R), Navigation and Support (N), Advertisement (A) and Form (F). Their algorithm automatically categorises Web elements by developing five graphs, one for each functional category, with the basic elements in the Web page as vertices. Each graph is specifically designed such that most of the probability of stationary distribution of a random walk is concentrated in the nodes that belong to its corresponding category. They are focusing on a very small set of roles/functions of web elements. However, in reality web elements have many more different roles.

- [3] uses heuristics to recognise the major sections of a web page which are: 1) top, 2) main content, 3) left and right menus, 4) bottom and 5) clutter such as advertisement.

  - Top includes the title of the page and a menu bar – Heuristic$_{top}$: "If a list of hyperlinks (i.e., a menu bar) or a table including a list of hyperlinks is placed within the top 200 pixels of the page [44], it is considered to be the top section."

  - Left and right menu includes navigation links – Heuristic$_{menu}$: "If a list of hyperlinks or a table including a list of hyperlinks is placed on the left (right) side of the page, occupying up to 30 percent of the page width [18], and its upper boundary is below the top section and its lower bound is above the bottom section, then it is considered to be the left (right) menu section.";
  - The main content is included in the center – Heuristic$_{main}$:"The remaining area (see Heuristic$_{top}$, Heuristic$_{bottom}$ and Heuristic$_{menu}$) is considered to be the main area";
  - Clutter usually contains images that are located in the bottom or side of a page.
  - Bottom – Heuristic$_{bottom}$: "If a table is placed within the lowest 150 pixels of the page [44], it is considered to be the bottom section.".

  [3] are very good and promising, however they are based on a specific model of web pages which is the header is at the top, the menus ate on the sides and the bottom of the page is clearly marked. Although this covers a wide variety of web pages, there are so many different structures on the web.

- [49] uses VIPS algorithm [18, 17] to first identify the blocks in a web page. By using heuristics these blocks are then categorised into three: Navigation bar, navigation list and content block. These three categories are then used to filter the unnecessary content (blocks which are not in these three categories). The following heuristics are used:

  - Navigation bar – Heuristic$_{bar}$: if the blocks have links, and if does not have another domain and if the over half of the content is links and if average link length is more than 10 and it does not have contents other than links then that block is a navigation bar.
  - Navigation list – Heuristic$_{list}$: if the blocks have links, and if does not have another domain and if the over half of the content is links and if average link length is more than 10 and it does have contents other than links then that block is a navigation list.
  - Contents – Heuristic$_{content}$: 1) If a block has links and the number of words is more than 100, then it is a content block. 2) If a block has links and it does not have another domain name and it does not have over half of links then it is a content block.

  It is again very difficult to see how these heuristics can be generalised, and also the role of the blocks are only focuses on these tree items which do not reflect the actual roles of elements in a web page. In summary, they do not really focus on identfigying the role of these blocks but they rather focus on grouping these blocks into three categories.

- [73, 74] differentiates link blocks from content blocks. Even though they do not try to understand the role of structural elements, their blocking algorithm does differentiate between content and link blocks.

- [26, 25] aims to identify high-level content blocks which are header, footer, sidebars (left and right) and the body. The authors also refer to these layout components altogether as semantic structure. They assume that header and footer typically has a flat shape, and header is located at the top and the footer is located at the bottom.

  - Header – Since the header block locates on the top of the page, they define a threshold N and let the upper N pixels of a web page to be the header region. They propose the formula: N=bas_threshold+F(heightwidth) where F(x) = a(b*x+c) where base_threshold, a,b and c are constants. Their experiments show that base_threshold=160, a=40, b=20 and c=1 are
  - Footer – Similar approach to header is used for the footer.
  - Left and Right Sidebar – 1/4 part of a web page to be the left sidebar and 1/4 of part of the page to be the right sidebar.

  Even though the main higher-level content blocks are covered, the way they are identified are focused on a specific web side template. Header at the top, footer at the bottom, left and right bars on each.

- [24] proposes a Function-Based Object model (FOM Model). According to this model, a web page consists of basic (smallest information body that cannot be further divided, only as a whole can perform certain functions) and composite objects (is a set of objects (both basic and composite) that perform certain functions together). They also propose a number of object categories:

  1. Information object, for content information
  2. Navigation object, provides navigation guide. Further divided as 1) navigation bar (provides global navigation), 2) navigation list (provides local navigation), 3) independent navigation guide (to provide navigation guide to certain piece of information)
  3. Interaction object, provides user side interaction
  4. Decoration object, serves only decoration purpose
  5. Special function object, performs special function such as advert, logo, contact, copyright, reference, etc.
  6. Page object, serves as the basic document of a web site, can be index page or content page.

  The authors also further define a number of rules for basic objects which include: 1) presentation property (media type, encoding formatting, layout information), 2) semanteme property, 3) navigation property (destination of a hyperlink), 4) decoration property (e.g, background colour), 5) interaction property (button, input, etc). Even though the proposed model is very comprehensive, it is very technical and does not take into account the users' understanding of the content. It does not also capture the detailed understanding of the document. For example, how would you categorise header, footer, etc?

- [44] aims to segment a web page into a set of coherent objects. The overall aim is to build a representation for a web page in which objects are placed into well-defined tree hierarchy according to where they belong in an HTML structure of a web page. They then focus on recognising some common areas on web pages such as header, footer, left and right menus, and the center of the page. They mainly define a set of heuristics to identify these common areas. These heuristics are based on a model where there is a rigid abstraction of the visual representation of the page.

- This is not so much related to classification of web page segments but it is about the classification of images [33]. By analysing images from different kinds of pages such as business, governmental, education, news, etc. and they propose the following three categories: 1) unlisted images: which are standalone images that appear anywhere in the page; 2) listed images: are two or more images that are systematically ordered with a web page; and 3) semi-listed images are visually similar to listed images. These groups are also differentiated based on their DOM tree.

- [70] defines a web page as a composition of basic visual blocks and separators – they indicate that visual blocks are visual parts in a web page that cannot be divided further. They have a very simply classification of blocks. They mainly classify them as nontext blocks (buttons, images, inputs, etc) and text blocks (is the area containing a paragraph of text, except text on forms). This is a very technical classification of content and has nothing to do with the role of the objects.

- [48] defines two types of blocks: informative and redundant content blocks. Informative content blocks include content which is semantically meaningful to users and redundant content blocks include redundant data such as company logos, navigation panels, advertisement banners, etc. Their focus is information retrieval therefore they focus on eliminating the intra-page redundancy.

- [11, 12, 13] indicates that web pages includes a lot of additional information besides the main content such as advertisements, copyright notices, etc that would effect the quality of data mining. Therefore, they propose an algorithm to identify the segments and based on the features of these segments to identify the main content. They mainly focus on identfying the following classes of objects: h1 (main artcile heading), h2 (second-level heading in the article), subtitle (the subtitle of the page), perex (the leading parag of the article), paragraph (an ordinary paragraph), data (publication date), author (author name), authordate (other object that belongs to the article), aobject (other object that belongs to the article), and none. Even though some of these are related to the role of segments in the page, it seems like they are mixed – some are related to the role of blocks in the page for example, h1, h2 etc and some are related to the semantics of the content for example author, authordate, etc. This is mainly because they are related to data mining. Furthermore, these are not systematically classified, they look like an add-hoc classification of the role of elements.

- [77] aims to eliminate blocks that contain noisy information with respect to data mining. Even though they do refer to different content blocks for example navigation panels, copyright, privacy notes, etc, they do not aim to identify the role of information blocks in a web page.

- [6, 63] aim to identify the fragments in a web page such that the page can be transcoded to better support accessibility for blind users. [6, 63] proposes different roles for these fragments which include: proper content, updated index, general index, no-role, header, footer, advertisement, delete, layouttable. Even though this looks like a comprehensive list, there still some roles that are missing in this group of roles. They have also not done systematically, it seems like the authors came up with a number of roles that are important from accessibility perspective.

3.1 The Importance of the Segments

There are also some work that aims to identify the importance of blocks/segments of web pages:

- [61] uses VIPS algorithm [18] to segment a web page and then proposes two learning algorithms to be used for identifying importance of these blocks. Block importances are originally identified with four levels: *level-1:* noisy information such as advertisement, copyright, decoration; *level-2:* useful information, but not very relevant to the topic of the page, such as navigation, directory, etc; *level-3:* relevant information to the theme of the page, but not with prominent importance, such as related topics, topic index, etc.; *level-4:* the most prominent part of the page such as headlines, main content, etc. However, after their experiment they proposed to have the following levels [61, 75] which means combining the second and third group: *level-1:* noisy information such as ads, copyright, decoration, etc.; *level-2:* Useful information, but not very relevant to the topic of a page, such as navigation, directory, etc. or relevant information to the theme of a page, but not with prominent importance, such as related topics, topic index, etc.; *level-3:* The most prominent part of a page, such as headlines, main content, etc.
- [48] aims to discover content blocks and in order to find the important blocks. They classify the blocks into redundant and informative content blocks.
- [11, 12, 13] aims to also identify the main content block. Therefore, they implicitly consider the importance of the blocks.
- [6, 63] also aims to annotate the fragments in a web page and their importance. The importance of a fragment is assigned with a real number which is between -1 and 1, 0 is the default role. The importance of the block is then used when the page is transcoded. The annotator can also specify the importance of a fragment and based on the importance value when the page is transcoded the fragment is located respectively. For example, if a fragment is annotated as an advertisement, then it receives an importance value which is -0.8 and then when the page is transcoded it is placed at the bottom of the page.

4 Why did it happen? Motivation

Web page segmentation has been done to address a problem in different fields including mobile web, archiving, phishing, etc. In this section, we summarise the problems that web

page segmentation addresses in different fields.

Mobile Web There are a number of problems associated with browsing web pages on mobile devices. These problems include the following: 1. the wireless bandwidth is quite limited and very expensive; 2. the screen size varies and can be very small; 3. some small screen devices have very limited memory and processing capabilities [75]; 4. the content of a large web page cannot fit into the memory of a device; 5. colour schema can be limited on a small screen device; 6. mobile devices do not have keyboard or mouse [75]; 7. Scaling down for small screen devices or scaling up for large displays could be difficult with web pages that are optimised for certain size [54]; 8. the same content is served to all users who are interacting in different context with the mobile web [46]; 9. the small screen device users have to scroll down a lot to access the content (the worst case is the two-dimensional scrolling) [72]; 10. since the web page is not properly marked up, it is very hard to personalise or customise pages for mobile devices [73]. 11. most small screen devices cannot handle multimedia data [42, 68]. 12. most web pages are designed to be displayed on large screens for Desktop machines [39, 78]. 13. the information need is very different for mobile web users, people focus more on getting direct answers to specific questions, and expect more relevant and clear results instead of browsing through large amount of data [75]. 14. context is also very important, for example location, personal profiles, time of the day, schedule, browsing history, etc. [75]. 15. For multi-platform development, the relationship between web page blocks and elements are not semantically explicitly specified so one cannot easily display a page on another or alternative platform [70, 78].

Voice browsers Voice browsers cannot directly access the segments in a web page to present them accordingly [56].

Web Page Phishing [1] defines phishing as a criminal trick of stealing one's personal information by sending them spoofed emails urging them to visit a forged website that looks like a true one. Identifying visual similarities between web pages is important for web page phishing detection [20, 67, 66, 50]. Only looking at the underlying HTML source code is not enough and it is important to investigate the visual similarities (the way users' see the web page) to detect phishing web pages [20].

Duplicate detection Duplicate web pages effect the user-experience and also consumes a lot of resources of search engines. Most duplicate detection algorithms are based on a concept called shingles [21]. Shingles are extracted by moving a fixed-sized window over the text of a web page and smallest shingles (in terms of hash values) are stored as the signature of a web page. These shingles are then used to compare web pages to identify duplicates. This approach is problematic because the noise in the web pages are also considered – there might be pages with same content but different noisy content. Therefore, [21] proposes to use segmentation algorithms to identify the main content and then take the signature of web pages for comparison purposes. [43] focuses on identifying identical content presented using different web page layouts.

Information Retrieval Traditionally, link analysis assume that a link represent a relationship between two web pages (A relates to B), however a link might represent a relationship between parts of web pages (part of A relates to part of B). Furthermore, noise in a web page can cause topic drift problem [18, 17]. [15] indicates that all link analysis algorithms are based on two assumptions: 1) the links convey human endorsement (if somebody adds a link to another page that means they endorse the other page) and 2) pages that are co-cited by a certain page are likely related to the same topic. Even though these are valid when the page includes a single topic, unfortunately most pages these days include a lot of sub-pages with different topics, for example the home page of Yahoo, MSN, etc. [19, 80] indicate two major issues: 1) if a web page is considered as a single semantic unit then it does not consider multiple topics in a web page, which means topics can be scattered at various regions of the page which can cause low retrieval precision; 2) if the page contains multiple unrelated topics then correlations among terms in a web page may be inappropriately calculated. [48] indicates that the redundant content in web pages such as advertisements, etc can be misleading for search engines. [48] indicates that search engines typically index whole text of a web page, and therefore they include a lot of text which is useless for processing, indexing, retrieving and extracting. Therefore, they aim to identify redundant content blocks such that the searching/information retrieval can be done in informative content blocks. [8] propose to identify pagelets in a web page to improve the performance of information retrieval. They mainly talk about two principles of information retrieval: 1) relevant linkage principle (web pages include many structural links such as menu links and these kinds of links violate the relevant linkage principle) and 2) topical unity principle (pages include mixture of topics which would negatively affect the information retrieval).

- Information retrieval - images (image search on the web): [40, 14, 65, 16] indicate that most search engines regard web pages as atomic units, however most web pages do not contain uniform information. They contain multiple topics with different kinds of segments such as navigation blocks, interaction elements, etc. Therefore, these affect the performance of image search algorithms on the web. [33] also indicates that traditional image search algorithms use fixed-length window size (for example, minimum 20 terms to maximum entire page) to extract the contextual information. [33] indicates that although this method is straightforward, this method tends to produce low-level accuracy as text tend to be associated with the wrong image.

Information Extraction To overcome the limitations of browsing and keyword searching, some researchers have been trying to build wrappers but for these wrappers block information is missing [18, 17]. [48] aims to identify informative and redundant content blocks, and then aims to make use of the features in the informative content blocks to support information extraction. [11, 12, 13] indicate that web pages include a lot of additional data and this data influences the result of the data mining algorithms, therefore it is important to identify the role and importance of web page components. [64] indicates that the source code of an HTML document typically focus on formatting the content of an HTML document, rather than semantically describing the content. Therefore, information extraction becomes challenging and the IE tools

adapt heuristics or other methods to interpret the content of a web page. [77] indicates that that the segments in a web page that contains irrelevant information to the main content of the page can easily harm the results of data mining. There are also many papers published on information extraction that focuses on record extraction [30, 47, 29, 57, 10, 62, 45, 31, 36, 28, 38, 5, 23, 32, 41]. They mainly focus on extracting unstructured data from web pages into databases where structured information is stored. Even though they are also related to web pages segmentation their focus is different. They mainly focus on extracting specific kind of data, for example weather forecast data, or prices, etc., however the segmentation algorithms that we refer here focus on segmenting a web page into a coherent sets of blocks. [30] defines a record as "a group of information relevant to some entity", here the focus is on the content. For example, a set of records can exist in a block.

User Interest Detection / Sensemaking Task [51] indicates that web site developers usually include different topics on web pages and when the user browses a page he is usually interested in only a part of it. When they browse that page, they are usually interested in the updates of that block and if that updated is automatically shown to the user, that will save significant amount of time of browsing time and it will improve user's browsing experience. [27] defines sensemaking as "sensemaking tasks require that users gather and comprehend information from many sources to answer complex questions". [27] indicates that even though search engines are targeting and helping users to find relevant resources the users still spend significant amount of time to find the information they are looking for in a given web page. They indicate that people spend 75% of their time in post-search phase of looking through individual web pages or sites. Therefore, web page segmentation tasks would be useful to highlight the relevant part to the user.

Evaluating Visual Quality (Aesthetics) [69] indicates that visual layout is one of the features that affect the visual quality of web pages which has been neglected in the literature. In order to process the visual layout, they view a web page as a semi-structured image.

Web page classification / Clustering [44] indicates that there might be noise in the page and some links that are located in those noisy parts of the page can be misleading for the classification of pages. They also indicate that one can suppose that words that belong to the central part of the page carry more information than words from the down right corner. Therefore, there should be a way to weigh words differently in different layout contexts. [77] indicates that the blocks in a web page that does not include content related to the main content can easily harm the data mining tasks such as web page classification and clustering. [77] aims to eliminate the noisy blocks from a web page to improve the performance of data mining tasks such as web page classification and categorisation.

Caching [58, 59] indicates that considering fragments of web pages for caching proved to be significant benefits for caching. Especially for dynamic web pages, identifying fragments that are shared between pages and have different life time can improve the efficiency of caching.

Archiving [60] indicates that in order to maintain a web archive up-to-date, crawlers harvest the web by iteratively downloading new versions of documents, however most of the time they retrieve pages with unimportant changes such as advertisements which are continually updated. Hence, web archive systems waste time and space for indexing and storing useless page versions. Furthermore, they indicate that querying such archive can be expensive because of the large amount of useless data stored.

Publishing Dynamic Content [22] proposes to use fragments instead of a whole page so that it is easier to update a page dynamically. Instead of having a complete web page archived and updated, they propose to compose a web page with fragments and then that would make the updates much easier.

Semantic annotation [55] indicates that most web pages on the web are encoded for human consumption and they are not designed for machine processing. [55] aims to identify segments that correspond to semantic concepts. In this respect, the overall idea is not about segmentation of a web page into cohesive number of blocks, but rather identify blocks that include semantic concepts.

Web accessibility When the page is visually fragmented and this is not encoded in the source code, applications such as screen readers cannot access that information. Therefore, the page becomes very inaccessible to screen reader users. [6, 63] aims to identify visually fragmented groups of elements in a web page such that the page can be transcoded to better support accessibility for blind users (screen reader users). They propose to manually annotate the page to identify the role of fragments in a page. [53, 9, 52] indicates that the applications such as screen readers process a web page sequentially (i.e., they first read through menus, banners, commercials, etc) therefore this makes browsing time-consuming and strenuous for screen reader users. Therefore, with the specialised audio browser called Hearsay or CSurf what they try to do is that when the user clicks on a link they aim to retrieve the target page and point the user to the relevant block to that link. They do this by segmenting a web page into a number of blocks and then identifying the context and the relevant block to that link.

5 Who is it about? Purpose

Based on what they want to achieve, different algorithms are developed for different purposes for different groups of users. Here we discuss the existing work in literature and what they do with the segments identified in a web page.

Mobile Web, After segments are identified in a web page, they are used for different purposes which can be summarised as follows:

- [78] aims to extract and present only the important parts of the web page for delivery to mobile devices. [61] similarly aims to discover the importance of segments of a web page which can be used to better adapt a web page for mobile devices – block-importance information can be used to decide which part of the page need to be displayed first to the user.

- Identify the main content of the page so that it is directly presented to the user [79], similarly, identifies the most important block/segment in the page and displays that to the user [75].
- Split a web page into a number of pages [3, 73, 26, 25]. [3] aims to identify information blocks so that a web page can be divided into smaller pages and represented to the user with a table of contents. [26, 25] proposes splitting the pages in two different ways: 1) single-subject splitting (one subpage is connected to another subpage) and 2)multi-subject splitting (one page is connected to a number of pages). [42, 68] proposes a transcoding technique called the indexed segmentation which transform segments or components of web pages into a sequence of small sub-pages that fit the display of a hand-held device, and binds them with hyperlinks. [39] proposes to segment the web page into a number of smaller pages and then displays a table of contents called "object lists" and provides a link to each segment from this table of contents.
- Zoom in and out of a web page or scale up or down a web page [54].
- Identify the navigation and content blocks, and then filter the relevant ones and show the summarises to the user[3].
- Provide a better navigation approach which would allow user to traverse the page from sub-page to sub-page [37].
- Present the complete web page to the user and then the user can click on regions on the web page to retrieve that sub-page (tap and display model) [72]. Similarly, [74] aims to show the user the thumbnail view of the page and when the users move their mouse over a section, they asynchronously retrieves that block and displays it to the user. Similarly, [26, 25] also aims to present the thumbnail view of the page and then the user can move to sub-pages. [76, 71] presents the thumbnail view of the page and the full screen information about a block, the user can move from one block to another to retrieve the content. [7] show the thumbnail view of the page to the user by dividing the page into 3x3 blocks, the user can then choose to further explore a block. [75] shows a thumbnail view of the page with the importance of the blocks/segments highlighted with different colours.
- Identify the blocks and filter out the irrelevant or unnecessary ones [73]. [24] similarly proposes to remove decoration and special objects such as adverts, logo, contact, copyright, reference, etc.
- Create a summary: [42, 68] proposes a transcoding technique for outlining the sections by using the section headers. In summary, the page is kept the same but the section header is converted to a link that points to the content of that section. This way the page is summarised.
- Create a summary: The content of the each block is removed from the page, and the first sentences of each block is turned into a link to the main content of that block. This way the page is summarised.
- Web adaptation in general, some work mainly focus on understanding of the structure to be able to adapt the content [34, 24, 18].

- Identify the navigation bar and do not display it if the page is a content page [24].
- Highlights the importance of blocks/segments in a web page with different colour. It shows a thumbnail view of the page with the importance of the blocks highlighted [75].

Detecting Phishing web Pages, [20] takes an image of a web page and aims to identify the blocks in a web page to detect the visual similarity between web pages for identifying phishing web pages. [67, 66, 50] aims to segment a web page to identify blocks and then compare them for similarity, similarity above a threshold is accepted as phishing pages.

Duplicate Detection, [21] aims to segment a web page so that the noise can be eliminated from the page to identify the main content to generate the signature of the page. [43] aims to segment web pages based on the text density to identify pages with the same content but different layout.

Web accessibility, [35] proposes to segment web pages for web accessibility or transaction management. [56] proposes a technique to segment web pages so that the structure can easily be accessed by voice browsers.

Information retrieval, [18, 17] proposes to use the segments to improve the precision of information retrieval. First, they retrieve the initial list of ranked web pages by using traditional information retrieval methods, they then apply their segmentation algorithm to the top 80 retrieved pages and get the candidate segments. They then choose the most relevant segments (for example, top 20) to chose the expansion terms. These selected terms are then used to construct a new expanded query to retrieve the final results. [15] uses the VIPS algorithm to identify the blocks, and then they extract page-block and block-page relationships, and construct a page graph and a block graph. This graph is then used in their two proposed information retrieval algorithms called Block-Level Page Rank (BLRP) and Block-Level HITS algorithm. [19, 80] evaluates the effect of four different segmentation method (fixed-length segmentation, DOM-based page segmentation, vision-based segmentation and a combined method) on the precision of information retrieval. [61] aims to segment the web page and discover the importance of these segments. This is mainly because during a search the most important part of the page can be treated more important which would give better search performance. [48] aims to identify informative content blocks and differentiate them from redundant content blocks. Their overall goal is to focus search on informative content blocks. Similarly, [8] aims to identify pagelets in a web page to improve the performance of information retrieval.

Mobile web search, [75] aims to use VIPS for segmenting web pages and discovering the importance of segments to better display pages for mobile devices. A simple search page is displayed to the user and when search results are returned, the returned pages are displayed in three different format: 1) highlights the importance of blocks with different colours, 2) creates a single column view of the segments and 3) displays the most important block [75].

Image search on the web, [40, 14, 65, 16] proposes to use VIPS algorithm to identify blocks in a web page to improve the image search on the web. Once blocks

are identified, they focus on image blocks (blocks that contain image) and then the content of those blocks are used to facilitate the search of images, for example, links, images, text in a block are used to provide better context to an image. [33] aims to identify segments to extract better contextual information for images – proposes a DOM-tree based segmentation algoritm.

Web page classification / clustering, [61] indicates that blocks and their importance can be identified, then when pages are compared, the features in an important block of a web page can be given higher weight. [77] aims to eliminate noisy blocks from a web page so that the pages can be better classified and categorised.

User Interest Detection, Sensemaking task [51] aims to find the block that the user is interested in so that the updates of that block can be automatically shown to the user. [27] proposes a tool that is browser extension to support sensemaking tasks. They mainly use a segmentation algorithm to identify the most relevant block to the users' current task by using the contextual information. Their algorithm dynamically analyses the pages that a user visits to determine the most relevant fragment of the content.

Evaluating Visual Quality (Aesthetics) [69] indicates that visual layout is one of the features that affect the visual quality of web pages which has been neglected in the literature. In order to process the visual layout, they view a web page as a semi-structured image.

Web page classification [44] indicates that one can suppose that words that belong to the central part of the page carry more information than words from the down right corner. Therefore, there should be a way to weigh words differently in different layout contexts.

Caching [58, 59] indicates that fragments that are shared between pages and fragments with different frequency of updates can be used for improving the efficiency of caching. These two types of fragments are used to decide when to and how cache the fragments of web pages.

Archiving [60] aims to segment a web page into blocks to be able to know accuratly when and how often important changes between versions occur in order to efficiently archive web pages. [60] uses VIPS algorithm to segment a web page and the proposes two algorithms one for detecting changes and one for identifying the importance of changes. Based on the importance they then decide to either update the archive or not to update the archive.

Information extraction [11, 12, 13] aims to identify the role of the web page segments to be able to support data mining. They indicate that the additional information on web pages influences the results of data mining algorithms. They focus on segmenting the page, identfying the role of these segments, and then focusing on the one that actually worths reading or that is to say worths considering for data mining. [64] aims to identify the interesting block (data block in a web page).

Semantic annotation / semantic web [55] indicates that most web pages on the web are encoded for human consumption and they are not designed for machine processing. [55] aims to identify segments that correspond to semantic concepts. In this respect, the overall idea is not about segmentation of a web page into cohesive number of blocks, but rather identify blocks that include semantic concepts.

Web accessibility [6, 63] aims to identify fragments and their roles such that the page can be transcoded to better support accessibility. In this work, the annotation is done manually. [53, 9, 52] indicates that the applications such as screen readers process a web page sequentially (i.e., they first read through menus, banners, commercials, etc) therefore this makes browsing time-consuming and strenuous for screen reader users. Therefore, with the specialised audio browser called Hearsay or CSurf what they try to do is that when the user clicks on a link they aim to retrieve the target page and point the user to the relevant block to that link. They do this by segmenting a web page into a number of blocks and then identifying the context and the relevant block to that link.

Publishing Dynamic Content [22] proposes to use fragments instead of a whole page so that it is easier to update a page dynamically. When an update occurs instead of updating the whole page, they propose a mechanism where the fragment of a page is updated.

6 How did it happen? Algorithms

For web page segmentation, different algorithms are proposed. There are mainly two kinds of approaches: top-down page segmentation vs bottom-up page segmentation [4]: Top-down approach start with the complete web page as a block and participation this block iteratively into smaller blocks using different features obtained from the content of the page [7, 18, 25, 39, 43]. Bottom-up approach for web page segmentation, mostly the leaf nodes of the DOM representation are taken as atomic content units [21, 43, **?**]. In this section, we summarise these algorithms.

1. [79] proposes an algorithm that automatically categorises the web elements. Their algorithm is based on random walks on specially designed graphs. For each web page, they develop five graphs, one for each functional category. Their work needs a training set that means the types of web pages they can work with is very limited. They cannot take any random web page and classify the elements of that web page. [73] proposes an algorithm based on DOM tree that aims to identify two **types of blocks**: link and content blocks. Their algorithm mainly focuses on Table tags. [26, 25] proposes an algorithm that first identifies the **higher-level content blocks** such as header, footer, sidebars and body. They then use two approaches to further divide the main content into a number of blocks: 1) explicit separator detection: they use three types of separators: HR tag, border properties of TABLE, TD and DIV elements, and images (by checking the width and the height of images); 2) implicit separator detection: implicit blank areas created intentionally by the author to separate the content.

2. [78] proposes a **ranking algorithm** similar to Google's pageRank algorithm to rank the content objects within a web page.They assume that the manner in which a person reads a web page is similar to how a surfer surfs the web. The reader enters the page through a link and is drawn to the elements that are related to the anchor text in the link located in central position of the page. The overall idea is to represent the web page as a graph and then exploit the graph structure of a web page to rank the elements. They first divide the web page into inseparable basic objects. They assume that the user is actually entering a web page from a link with an anchor text. They then rank the relevance of each inseparable object to the text of the link anchor. Based on these rankings and also the relationship between basic elements, they form a semantic graph. This semantic graph is then used to select a rectangle covering all the important elements of the web page and this rectangle is then transmitted to the user. **Limitation**: Their algorithm only works for pages that the user is traversing a link. Their algorithm does not work if the user is randomly visiting a page.

3. [3] have a set of **heuristics** that they use to first identify the main content of the page (they have a very simple model of the page where it consists of top, menus, main content and bottom part) and then they have a set of heuristics to identify the blocks (atomic and composite) blocks in the main content. HTML elements are mainly grouped into four groups: structure, formatting, header and separator, and these four groups are then used to devise the set of heuristics for identifying the blocks in the main content. [54] proposes to use a number of **heuristics** such as tables to break the page into logical units. [44] uses a number of heuristics to identify the common areas of web pages. They use both the underlying DOM and also the visual rendering or coordinates to identify these pages. They have predefined coordinates for position of areas of interest in a page. Proposed heuristics are based on this predefined model. [11, 12, 13] also uses heuristics in their algorithm. Their algorithm has four steps: 1. page rendering in order to obtain the visual rendering information of objects, 2. detecting basic visual blocks in order to obtain tree of areas that show the blocks in the page, 3. text line detection to join the same areas in the same line and 4. block detection to detect the larger areas with the same visual style of blocks.

4. [18, 17, **?**, 19, 80] proposes the VIPs algorithm. VIPS uses both the DOM and the visual presentation of the page. VIPS algorithm has three steps: 1) block extraction (a web page is recursively divided into blocks by using a number of heuristics), 2) separator detection (separators are horizontal and vertical lines in a web page that visually cross with no blocks in the pool) and 3) content structure construction (this step is used to construct the structure tree of the page). [3] uses the **VIPS algorithm** to identify the blocks in a web page. They then use heuristics to groups these blocks into there categories: navigation list, navigation bar and contents. The blocks that do not belong to these categories are then filtered out, and the rest of the blocks are summarised and personalised to the user. [72] uses the VIPS algorithm to identify the blocks in a web page. They mainly display the complete web page to the user and the user is able to click on the regions of the page to retrieve the relevant sub-page. [74] also uses VIPS to identify the blocks. [61] uses VIPS algorithm to segment a web page and then it uses two different learning algorithm (SVM, neural network) to discover the importance of blocks based on their spatial features (such as position,

size) and their content features (such as no. of images). Application of VIPS include:

- identifying importance of blocks [61];
- block-based web search [18, 17, 80, 15, 19]
- web search on mobile devices, [75] uses VIPS algorithm to segment a web page and proposes three different presentation techniques based on these segments which include: 1) displaying a web page with importance of segments highlighted, 2) ordering the segments into single column view of the page and 3) identifying the most important block and displaying that to the user.
- image search on the web, [40, 14, 65, 16] propose to use VIPS algorithm to identify blocks in a web page to improve the image search on the web. Once blocks are identified, they focus on image blocks (blocks that contain image) and then the content of those blocks are used to facilitate the search of images, for example, links, images, text in a block are used to provide better context to an image.
- Evaluating visual quality or aesthetics of a web page, [69] uses VIPS to segment a web page into a number of blocks and then use this information in the algorithm they propose for evaluating the quality of a web page in terms of aesthetics. They mainly extend VIPS and they called it VIPS based layout block extraction algorithm (V-LBE). This algorithm mainly selects the block candidates whose sizes are above a threshold and deletes or inserts blocks to construct a set of un-overlapping large blocks.
- [60] uses VIPs algorithm to identify if the change in the page is important for archiving.

5. [37] proposes an algorithm to identify sub-pages which is mainly based on the **size of the screen** width and the height. Compared to other proposed system, this mainly takes into account the device capability for breaking up the page or identifying the sub-pages. [35] proposes two algorithms: 1) an algorithm to find blocks (geometrically aligned blocks are identified) and 2) an algorithm to find segments (uses presentation information to segment a block into a number of partitions). [39, 27] proposes an algorithm that segments a web page based on the distance between elements. In the original algorithm they have only used DOM to calculate the distance between elements to segment the page into two-three segments, but in the further work they have also used the layout information for segmentation. [34] proposes a system that includes mainly two operations; dividing (divides the page into as many segments as possible) and merging (combines some segments based on their visual similarity). [24] uses the properties of objects as they have described in their model and then they analyse a web page with these properties in mind to identify the objects – they mainly convert the HTML documents into a number of WAP decks. [51] proposes an algorithm based on DOM which has three steps. First they remove the null nodes (nodes that do not contain any spatial information – nodes that do not occupy any space on the page and nodes that have some children but their space is not larger than its children nodes) and then they try to detect the separators between nodes. A separator node is defined as a node that doesn't have any child nodes, images that have a

size smaller than threshold. The last step is to traverse the compact DOM to identify the blocks. [70] proposes that a web page is considered as a composition of basic visual blocks and seperators. Therefore, their algorithm focuses on first identifying the blocks and then discovering the separators between these blocks. [64] focuses on two types of nodes features to identify segments in a web page: node content size (text manifested by its subtree in the rendered web page – number of words of text in its contents) and node entropy (identify the patterns in a node). [77] proposes to create a style tree of a web page and then based on this style, they propose to identify which parts are noisy and which parts represent the main content of a web page. [8] proposes a page partitioning algorithm. This algorithm mainly applies the following definition of a pagelet "An HTML element in the parse tree of a page p is a pagelet if (1) none of its children contains at least k hyperlinks; and (2) none of its ancestor element is a pagelet".

6. [43] proposes an algorithm that segments a web page by focusing on the text density.

7. [56] proposes an algorithm that aims to identify the page segments by using the **uniformity** of the web elements, particularly $\prec$a$\succ$ tag. Their approach has three steps: 1) preprocessing, 2) segmentation and structuring and 3) post processing. In the processing part, the document is converted to XML, in the second part based on the uniformity of the elements, segments are recursively identified and in the last part focuses on grouping the elements that cannot be successfully processed in the first two steps.

8. [20] takes an image of a web page and then aims to identify blocks. Their algorithm works on the following two assumptions: (a) a web page is composed of blocks, every visible element in HTML is displayed in its rectangle area. Labels, images, tables even flashes are all possessed of the basic parameters: width and length. So with a rectangle area, we can locate any visible element in web pages (b) the visible elements are separated by background space. If there is no space between two visible elements, they may be considered as one by people, which does not handicap our understanding from visual aspect. Their algorithm works well if the page satisfies these two conditions.

9. [42, 68] proposes a technique for identifying blocks in a web page by investigating the repeated patterns. They mainly take the HTML tags and convert them to a pattern in string. They then look for repeated patterns.

10. [76, 71] proposes a segmentation algorithm based on the Gestalt theory which is a psychology theory to explain human's perceptive process. According to this theory humans visually recognise a figure or form as whole instead of just a collection of simple points, lines and curves. According to gestalt theory human's perceptive process is guided by four general laws which are proximity (items tend to be grouped together according to their nearness), similarity (similar items tend to be grouped together), closure (items are grouped together if they tend to complete some structure) and simplicity (items tend to be organised into simple structures according to symmetry, regularity and smoothness).

11. [7] proposes a machine learning algorithm that segments a web page into coherent regions based on clues from the DOM combined with simple computer vision algorithms. Their algorithm is mainly a decision tree based segmentation algorithm. They focus on dividing the page into 9 blocks (3x3), and allow the user to be able to focus on each block to further read the content. [21] also proposes a machine learning algorithm. Their approach is based on a combinatorial optimisation framework. They particularly cast it as a minimisation problem on a suitably defined weighted graph, whose nodes are the DOM nodes and the edge-weights express the cost of placing the end points in same/different segments. They then take this abstract formulation and produce two concrete instantiations, one based on correlation clustering and another based on energy-minimizing cuts in graphs. [53, 9, 52] also proposes an algorithm based on Support Vector Machine which is a statistical machine learning algorithm.

12. [4] proposes an algorithm for web page segmentation which is based on clustering web content. They define web content as the smallest indivisible content of a web page which is usually represented as a leaf node in the DOM tree. They also refer to block as web contents that share a coherent topic. style or structure. Therefore, they define web page segmentation as a grouping of web contents into blocks. In their algorithm they use both DOM-based distance and geometric distance in the visual rendering of web pages.

13. [48] proposes an algorithm that focuses on tables for segmenting web pages. They mainly assume that tables are used for structuring web pages.

6.1 Algorithmic Assumptions

Some of these algorithms make key assumptions:

- The reader enters the page through a link and is drawn to the elements that are related to the anchor text in the link located in central position of the page [78].
- [35] assumes that the semantically related pieces of content in a web page tend to be located near each other, often sharing the same alignment.
- A web page is consists of five components: top, main, left and right menu and bottom part. Although this is a useful model to simplify the structure of a web page, it can hardly be generalised [3].
- Uniformity of certain information is the key in understanding the segment and structure [56].
- When perceiving a web page, human unconsciously follow the four laws of gestalt theory and segment it into several hierarchical parts by integrating various cues presented in the page [76, 71]. However, there is no scientific evidence that that is the case.
- Some algorithms assume that the thumbnail view of the page is useful on mobile devices [7].

- When a designer creates a web page they actually group the content of the same topic in one block [51].
- Certain HTML elements are used for certain purposes. For example, [48] assumes that tables are used for structuring purposes.
- [55] proposes an algorithm based on the following two observations: semantically related items exhibit consistency in presentation style and semantically related items exhibit spatial locality. Even though these observations are important they are mainly focusing on the semantics of the segments rather than the information blocks presented.

6.2 Limitations of the Proposed Algorithms

Some of these algorithms/solutions have very important limitations:

- Their algorithm only works for pages that the user is traversing a link. Their algorithm does not work if the user is randomly visiting a page [78].
- The algorithm needs to be trained [78, 79].
- Heuristics developed are based on simple models that cannot be generalised, for example web page is consists of five components: top, main, left and right menu and bottom part. Although this is a useful model to simplify the structure of a web page, it can hardly be generalised [3].
- The main input to the algorithm is the HTML document, in that case the information about the overall structure of the page is missing [56].
- Most web documents are not properly formatted, some algorithms focus on using tools like Tidy to generate documents that can be properly processed [56].
- Some algorithms focus on certain HTML elements, for example [73] focuses on using Table elements. Even though tables provide a lot of information about the underlying structure of the page, focusing on a specific element is not good for the generalisibility.
- Some algorithms only focus on certain web page templates, for example [26, 25] focuses on pages that have the following structure: header at the top, footer at the bottom, sidebar on the right and on the left of the page, and the content is in the middle. Even though such assumptions are important for the success of the algorithm, again it is hard to see how the proposed approach can be used for pages with other semantic structures.
- Some algorithms make assumptions about the way a web page is designed and laid out. For example, [20] assumes that a web page is composed by blocks and the visible elements of web pages are separated by background space. Although majority of web sites satisfy these, there are some cases where the layout does not fit into this definition.

- Some algorithms are based on pattern matching in HTML [42, 68]. Even though this is a simple and sounds like a good approach, it has a number of limitation. For example, there might be the cases where two visual blocks look exactly the same but they are coded differently, for instance one can code a block with DIV element or with a TABLE element. Therefore, a pattern matching with only HTML content can be very limited.

- When perceiving a web page, human unconsciously follow the four laws of gestalt theory and segment it into several hierarchical parts by integrating various cues presented in the page [76, 71]. However, there is no scientific evidence that that is the case.

- Some algorithms need to be trained and this could be an issue in the overall automation of the process [21].

- Some algorithms are focused on certain HTML elements, for example [48] assumes that web pages are structured by table elements.

7 Where did it happen? Server, Proxy or Client-Side

Based on the publication and delivery model of web pages, segmentation can be done in three different position of the delivery which are: server, proxy and client. This section discusses the different approaches taken for segmentation.

Server There are a number of *advantages* if the processing is done on the server side [2, 73, 26, 20, 15, 22], [20, 75] works offline on the server side. Image search has been done on the server side [40, 14, 65, 16].

- Content can easily processed on the server side;
- The content owner can control the way the content is processed;
- The owner can protect the copyright;
- This means the minimum network bandwidth;

There are also a number of *disadvantages* if the processing is done on the server side [2]:

- The engine has to be implemented on the server side along with the main content.
- Alternatively, alternative copy of the page has to be served.

Proxy [74, 46, 37, 72, 73, 74, 26, 25, 42, 68, 76, 71, 7, 39, 6, 63]. Proxy implementation has a number of *advantages*:

- This means that the application is independent of the content provider and the client;

- The proxy server will be dedicated for this task which would mean efficiency in processing;

There are also a number of *disadvantages* if the processing is done on the proxy server [2]:

- Availability of the proxy server to the client's terminal;
- The connection speed between the client and the proxy and between the proxy and the server serving the content is important;
- Copyright issues have to be considered for the content processed in the proxy.

Client [54, 79, 78, 35, 3, 56, 26, 21, 64, 55]. [3] proposes a specialised browser. [18, 17] proposes a specialised application which is based on Internet Explorer engine. [61] proposes a client application that can used to identify the importance of blocks in a web page. [53, 9, 52] propose a specialised audio browser called HearSay (later on with the contextual work they call it CSurf). It has a number of *advantages* if it is done on the client side. [27] proposes a browser extension sensemaking tool.

- The user can specify their preferences and determine the scale of the segmentation;

There are also a number of *disadvantages* if the processing is done on the client side [2]:

- The main limitation is the resources available on the client side – processing capabilities of devices, and memory limitations.
- The user needs to install a third-party application or a browser plug-inn;
- Multiple variants of the application has to be prepared for different devices which can be a tedious and complex task.

7.1 Input Used in the Proposed Algorithms

HTML elements (structure-based) [56] proposes an algorithm based on HTML tags.

DOM interface (structure-based) [79, 78, 37, 26, 25, 42, 68, 19, 80, 24, 51, 58, 59, 77, 55, 8, 6, 63] uses the DOM interface. [73] uses the DOM tree of the web page and also before the processing they remove the noise from the content such as annotation, scripts (CSS), images, etc. [33] uses the DOM tree for segmentation. [64] also uses DOM.
PROS

- Depth and type of the node in the DOM gives an indication of how much it should be kept together.
- May capture many of the benefits of the vision based techniques, as if there is a background that is different, it can appear in the DOM.
- it is simple and scalable [11, 12, 13].

CONS

- Examining only DOM elements may not capture obvious visual cues and may make distinctions between regions that appear similar.
- Further, this technique may require some basic computer vision technique to solve uncertainties of where to place cuts between DOM nodes.
- Some elements are very closely located in the HTML source code (for example, two cells in a table element) but they are visually displayed very far apart from each other in the visual rendering [39].
- [39] also indicates that most web sites adapt a layout that includes components such as header, footer, menu, etc and each usually have different structure but if only DOM is used then these different structures cannot be easily identified.
- [18, 17] indicates that since most people do not obey W3C specification, there can be a lot of mistakes in the DOM tree.
- Most authors also do not use DOM for encoding the semantic structure of the web page, for example two elements might have the same parent, but content wise (semantically) they might be very related.
- [18, 17] also indicates that even though XHTML is introduced as an XML extension of HTML which can be used for semantic encoding, not many people use it.
- [43] indicates that there are so many different ways to model an identical layout, for example using Table or div elements for blocking, using I or B tags or using CSS.
- [43] indicates that because of the heterogeneity of HTML style, algorithms are susceptible to failures.
- [11, 12, 13] indicates that the DOM needs to be updated with the additional information encoded in the external files such as javascripts and CSS files.

Visual Rendering (layout-based) [35] proposes to use the visual rendering of the web page provided by Mozilla. The authors indicate that their work is based on the following observation "Information about spatial locality is most often used to cluster, or draw boundaries around groups of items in web page, while information about presentation style similarity is used to segment or draw boundaries between groups of items". [34] focuses on the visual realisation of the page rather than the underlying code, their work is tag-independent. [70] uses only visual rendering of a web page from IE. [11, 12, 13] also proposes an algorithm that uses the visual rendering of the page.

PROS

- Better at keeping salient portions together, especially when the portions are noticeably different colours (for example, a region is highlighted or has a different background)

CONS

- Does not take into account the DOM of the web page, therefore no notion of which regions are more important to keep together, will not work on simple pages, if the regions are not visible different, thie method may not work as well.

- Vision based approaches naturally have a higher complexity since the layout must be rendered prior to analysis, which might be too slow to be incorporated into the web crawling and indexing cycles [43].

Hybrid approach (both DOM interface and visual rendering) [3] uses both structural and visual layout information of a web page to detect related content. [3, 72, 74] uses the VIPS algorithm. [76, 71] uses the DOM tree and the visual rendering of the page to create a feature tree by feature extraction and structural refining. [7] uses both DOM and the visual rendering of the page. [21] uses three sources of information: 1) each node in the tree represents a subtree that consists of stylistic and semantic properties; 2) each node occupies visual real-estate when rendered on a browser; 3) same visual layout can be obtained from syntactically different DOM trees. [39] proposes to use both the DOM and the layout of the page – it is interesting to see that their original work which was only using DOM was not good enough, had some problems and they show that their new proposed hybrid approach works better [39]. [18, 17, 15, 19, 80, 61, 75, 40, 14, 65, 16, 69, 44] uses both DOM and the visual rendering of the page.

Image [20] takes an image of a web page and does image processing for identifying the blocks visually in a web page.

Fixed-length segmentation [19] indicates that fixed-length is used to overcome the difficulty of length normalisation problem in traditional text retrieval. [19] removes all the semantic information, tags, from the page, and then uses a fixed-length to segment a web page.

Text-based segmentation [43] aims to retrieve segments from web pages based on the low-level properties of text instead of DOM-structural properties.

7.2 Limitations and Weaknesses of the Inputs Used

Based on what these algorithms use the following observations have been made:

1. The visual rendering of the web page contains a lot of implicit information about the content of the web page that cannot be accessed via the DOM tree of the web page. DOM trees contain only information about the location of web page elements relative to each other, they lose information about the visual layout that is useful for partitioning [35].

2. Only structure-based approaches suffer from unstructured web pages;

3. Only layout-template based approaches can cause false detection in layout-based approaches.

4. Some algorithms [73] also filter out some content that they call them the noise, such as annotation, images, scripts. Even though removing these would be good for processing efficiency, they are also part of the document and some of them are important for understanding the overall structure and interaction supported by the page, for example CSS is important for the layout, other scripts are important for the interaction and images are also part of the structure and information presented in the page.

5. Most of these algorithms cannot handle Java, Javascripts or other kinds of scripts, Flash content, etc. [42, 68].

6. There are a lot of web pages that are not properly structured HTML (pages with irregular HTML). [39] indicates that in their study of irregularly tagged HTML documents, 8.5-27.1% of pages have problems that cannot be automatically fixed with tools like Tidy. This percentage is quite high [39]. This would definitely an issue that would affect all automated algorithms. [44] similarly indicates that the HTML source codes are far from the standard and posed a lot of difficulties in rendering these pages properly. Therefore, they propose to simplify the rendering for example by ignoring frames. layers and stylesheets. Even though these simplifications mean it would be easier to process the document, that means the processed document is different from the original document and these differences can cause a lot of differences in the resulting fragmentation of the document.

7. [33] indicates that when visual information is also used in an algorithm for segmentation, this introduces extra computational expense, and becomes difficult to generalise it to all the pages on the web.

8 What happened? Evaluation

Precision and Recall [79] uses precision, recall and f-measure to check the validity of their proposed approach. They have manually labelled 12,134 elements from 150 web sites into one the of the proposed categories, they have then used 10,009 as the training set and 2,125 as the test set. They have then concluded that the random walks is an effective and practical method. [46] proposes to use three blog categories: navigation bar, navigation list and content blocks. In their evaluation, they have used 8 sites and 100 pages and tested whether the block filtering is performed correctly. [76, 71] also uses precision as the metric for comparing their algorithm with the VIPS algorithm. According to their paper, precision is defined as summed area of blocks/summed area of all blocks in the resulting set. They have also manually grouped the identified blocks as error, not bad and perfect, and they have investigated precision for each group. [39] used precision and recall to compare their advance algorithm with their original algorithm. They have mainly used the following two formulas to compute F-measure:

1. Precision: number of segments / number of all segments;
2. Recall: number of correct segments / number of all correct segments.

[65] also uses precision and recall to evaluate the proposed image search on the web. They have used 10 volunteers to manually label the ground truth. They have then investigated precision and recall to check the performance of the search algorithm based on segmentation of web pages. [70] also uses precision and recall – mainly compares automated vs. manual blocking. [48] also compares their algorithm with manually annotated web pages. Their algorithm focuses on identifying informative content

blocks and therefore they manually annotate informative content blocks and then they assume that the features extracted from these manually annotated blocks as desired features. They then define precision and recall as follows: recall rate=common features/desired features and precision rate=common features/discovered features, and they define the ideal case as precision and recall is equal to 1. [11, 12, 13] also uses precision and recall. They have first manually annotated 9 common areas in a web page and the computed them automatically and compared the relationship between these via precision and recall. [64] also uses precision and recall. [64] also uses precision and recall for evaluating the performance of the proposed algorithm.

Recall and The percentage of returned elements of the extraction [78] focuses on measuring two parameters: 1) The recall value R (retrieved elements that are relevant/all the retrieved elements) and 2) The percentage of the returned elements of the extraction (number of retrieved elements/number of elements on the web page). Their focus is to obtain a high recall value (R) and reduce the return rate (deliver as little content as possible). They have selected 158 websites from Google directory, under the category of news. They then chose 5 web pages and recorded the anchor text of links in these web pages. In their evaluation, they have then used 3 people to highlight (draw a rectangle) the part of the web page that they would like to read on a small screen device. Some of the data is used for training the application and some data used to evaluate the system. **Issues:** 1) small number of people are used to annotate the parts of the page that they considered as main content, 2) the web page used are very specific (news sites) as they have a very specific structure, 3) context is not considered – people might read different parts of the page in different context, without knowing the users' task it is very difficult to say which part of the page they want to actually read.

Success rate or Accuracy [3] proposes a set of heuristics to first identify common parts of pages including the main content and then proposes a set of heuristics to identify the topic blocks. The authors also present an evaluation where 20 web pages from three categories news, travel and shopping are used to identify the success rate of these heuristics. The pages are manually annotated and then then the coverage of heuristics are given as percentages. [73] evaluates 100 pages by looking at the generated block structures and classifying the generated blocks as successful and unsuccessful they than compute accuracy rate based on these results. [26, 25] collected 200 pages from 50 popular sites, which are processed and divided into blocks by the proposed algorithm. A number of testers are then asked to evaluate these by putting them into three categories: perfect (there is no error), good (analysis is correct but there are some errors in splitting) and error (there are errors in both analysis and splitting). [42, 68] also proposes an evaluation with 43 students to assess 13 web pages. They are mainly asked to rate the quality of transcoding as fair, good, excellent, usable and poor. [76, 71] also manually classifies blocks as error, not-bad, perfect. [21] compares the segments generated by their algorithm with the manually segmented pages. They have used 1088 segments from 105 web pages. They have used two metrics for comparing manually generated segments with automatically generated segments. These metrics are commonly used in machine learning literature to compute the accuracy of clusterings with respect to ground truth. These metrics are Adjusted Rand

Index which is a measure of clustering and Normalized Mutual Information which is used to showcase the difference between the two algorithms proposed by the author [21]. [18, 17] used 5 volunteers to judge 600 web pages segmented with their algorithm as perfect, satisfactory, fair, and bad. According to their results [18, 17] 93% of the experimented pages were segmented correctly. [51] proposes to track 12 popular web sites over a month where for each web site they have collected 25 copies of their home pages. They have then segmented these pages into blocks and then they manually look at their correctness. [44] first constructed a dataset that includes 1000 pages and then they had two people to manually label the segments as header, footer, left menu, right menu and main content. Their rating were based on the metrics: good, bad, excellent, and not recognized. They have then performed automated evaluation and compared the success rate of manual vs automated area detection.

Simulation [37] evaluates the proposed system with a simulator.

User evaluation - Simulation [39] evaluates the system by simulating how a user would interact with a segment web page. They have mainly calculated the estimated time to reach a part of a web page. They have compared their results with the Google Wireless Transcoder. They have chosen 5 web pages and then they have chosen imaginary targets from top/middle/bottom. They have then calculated estimated time to reach these imaginary targets and compared their results with Google Wireless Transcoder. [75] presents a user evaluation with 25 users and with 8 different search tasks. Even though this user evaluation shows promising results, it is not clear how the segmentation is evaluated here. The focus is more on the different presentation techniques and their effectiveness in terms of the search time and user experience. [51] first segments pages into blocks and then detects the block that the user is interested in. They have done user evaluation with four users regarding the interested block detection. Even though this user evaluation is important, it is not clear how the blocking is evaluated with this user evaluation. [27] aims to identify the block that is the most relevant block to the users' content and conducts two user evaluation: 1) one focuses on evaluating the accuracy of the identified unit, and 2) one focuses on the task completion time. These user evaluations again does not directly address the segmentation algorithm but it addresses both the segmentation algorithm and the proposed sensemaking tool.

Task-based user evaluation [72] performed a user evaluation where a number of users are asked to perform tasks with and without partitioning the page. However, with the user evaluation, it is not clear what the evaluation addresses, does it address the success of the partitioning or does it address the success of interaction model? it is very hard to observe the effect of both on the user evaluation. [27] aims to identify the block that is the most relevant block to the users' content and conducts two user evaluation: 1) one focuses on evaluating the accuracy of the identified unit, and 2) one focuses on the task completion time. These user evaluations again does not directly address the segmentation algorithm but it addresses both the segmentation algorithm and the proposed sensemaking tool. [6, 63] also performs evaluation but their evaluation focuses on the manual annotation aspect. Therefore, it is very hard to see if the proposed fragmentation and also the roles do improve the accessibility of a page.

Execution time or speed or output size [72] investigates the processing time in the proxy

for analysing the page. [73] evaluates the speed and execution time of their system with 3000 pages from 10 typical portal websites. Similar evaluation has also been presented in their follow on work [74]. They have also compared the speed of their new AJAX approach [74] with their original work [73]. [56] uses two metrics to check robustness: 1) successful processing – were the system able to process the given web pages and 2) time to complete the processing. They have also investigate the accuracy of segmentation and structuring. They have collected 70 pages and used three participants to rate the accuracy of the generated segments and structures between 0-5. [26, 25] iinvestigates user=perceived delay (PD) which includes page downloading time from server to proxy (DT), processing time at the proxy (PT), page-downloading time from proxy to client and rendering time at client. However, they have only measured DT and PT. [20] also investigates the speed of segmentation algorithm. [76, 71] investigates the average processing time of web pages from different domains including auto, bank, e-commerce, finance, IT info, media, music, news, search, sports, university, all. [39] examines the processing time of major activities of their proxy which includes: parsing HTML, rescaling images, extracting tag depth, segmentation and rebuilding XHTML. [60] uses the VIPs algorithm and evaluates the execution time and output size. Output size is important as they use the segmentation process as part of archiving.

Comparison of Algorithms [20] compares their phishing algorithm with another algorithm. Although this is useful for investigating if the proposed application of segmentation works, the segmentation itself is not directly evaluated. [76, 71] compares their segmentation algorithm with the VIPS algorithm [18, 17]. [40] compares their image search algorithm based on VIPS with another image search algorithm. [43] compares different algorithms that they propose and also they compare their algorithm against the algorithm proposed in [21]. [4] compares a number of variations in their algorithm. They also have three people participating in their data collection process. Even though three people can generate a lot of data, one can easily question the validity and objectivity of the proposed evaluation.

Small set testing [7] demonstrates that their algorithm works with a number of popular web pages. [18, 17] performed three different type of evaluation and one of them was testing a number of pages with their algorithm.

Information Retrieval Experiment [18, 17, 15] test their segmentation algorithm used in an information retrieval experiment. Even though it is good to see how the segmentation algorithm can be used and perform in an application, when an information retrieval experiment is performed, the question is how does the other parameters affect the overall experiment of the segmentation algorithm? [19, 80] tests the effect of different segmentation algorithms on information retrieval precision. [8] also tests their algorithm with information retrieval algorithms.

[77] evaluates their proposed algorithm with two data mining tasks: clustering and classification. They compare F scores with and without noise elimination.

8.1 Problems in the Proposed Evaluations

- Small number of people are used to annotate the parts of the page that they considered as main content, 3 people used to annotate the relevant content [78].
- Context is not considered – people might read different parts of the page in different context, without knowing the users' task it is very difficult to say which part of the page they want to actually read [78].
- The web page used are very specific (news sites) as they have a very specific structure [78].
- The evaluation is not done systematically and weak qualitative data is provided. For example, [3] presents a study of 20 web pages for the presented heuristics however the evaluation is not done systematically and it is not clear how the success rate is calculated. [46] presents an experiment with a small data set and the manual annotation is not explained in detail. It is not explained well how many testers are used to look at the automatically processed pages and grouped them either as perfect/good/error [26, 25] or as successful/unsuccessful [73]
- Although simulation could be a good way of evaluating the proposed algorithm, [37] does not do it systematically and they only show that their system works with a small data set.
- With a user evaluation, it is not clear what the evaluation addresses, does it address the success of the partitioning or does it address the success of interaction model? it is very hard to observe the effect of both on the user evaluation. Even though this user evaluation shows promising results, it is not clear how the segmentation is evaluated here. The focus is more on the different presentation techniques and their effectiveness in terms of the search time and user experience [75].
- [20] compares their phishing algorithm with another algorithm. Although this is useful for investigating if the proposed application of segmentation works, the segmentation itself is not directly evaluated. Obviously the segmentation algorithm affects the phishing results, the segmentation algorithm is not directly addressed. Similarly, [67, 66, 50] compares phishing pages with genuine pages, and also compares genuine pages with their algorithm. Even though this is good evaluation of the proposed phishing algorithm, it is not clear how the segmentation affects the results or the accuracy of their phishing algorithm.
- When processing time is investigated, the number of pages is very limited and it is not clear what kind of pages are used to test the processing time, for example did they use simple pages or complex pages? Did they use pages from different domain, for example [76, 71] covers wide variety of domain.
- When accuracy is investigated (i.e., a number of users are asked to rate generated blocks, for example as error, not-bad, perfect [76, 71]), either the number of testers are very low or it is not clear who has actually tested these generated blocks, were the authors involved [76, 71, 39]?

- Testing with a small set is not enough to show that the proposed algorithm works well, for example [7] shows that their algorithm works with a number of pages but it is not clear if the proposed system can scale or can work with different pages, etc.
- Even thougsh user evaluation with simulating user interaction is a good idea [39], one needs to be careful with the assumptions. For example, how do you decide what would be the target of the user? How would you decide what would be the context of the user? how would you decide if the user is alone or with somebody? etc. All these assumptions make the evaluation unrealistic.
- number of users are asked to rate the success of transcoding [42, 68], here the problem is again does the evaluation address the success of block identification or does the evaluation address the transcoding which is what you do with blocks afterwords? Similarly, [18, 17, 15] performs an information retrieval experiment with their segmentation algorithm, but it is not clear if this experiment addresses the success of information retrieval algorithm or does it address the success of segmentation?
- Most of these algorithms choose a number of pages for evaluating their algorithm, however it is not clear how these pages are selected. Most of the time, there is no systematic method for choosing the pages used in the evaluation [51]. This would definitely affect the generalisability and validity tests of these proposed algorithms.

8.2 Observations

- Combination of algorithms would be good and useful to overcome limitations of different approaches [43].
- Hybrid input data approach is also good for overcoming the limitations of different types of inputs [43].

9 Summary

Web pages are typically designed for visual interaction. In order to support visual interaction they are designed to include a number of visual segments. These visual segments typically include different kinds of information, for example they are used to segment a web page into a number of logical sections such as header, footer, menu, etc. They are also used to differentiate the presentation of different kinds of information. For example, on the news site they are used to differentiate different news items. This technical report aims to review what has been done in the literature to automatically identify such segments in a web page. This technical report reviews the state of the art segmentation algorithms. It reviews the literature with a systematic framework which aims to summarize the five Ws – the 'Who, What, Where, When, and Why' questions that need to be addressed to understand roles of web page segmentation.

References

[1] Anti-phishing working group. `http://www.antiphishing.org/`.

[2] Velibor Adzic, Hari Kalva, and Borko Furht. A survey of multimedia content adaptation for mobile devices. *Multimedia Tools and Applications*, 51:379–396, 2011. 10.1007/s11042-010-0669-x.

[3] Hamed Ahmadi and Jun Kong. Efficient web browsing on small screens. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 23–30, New York, NY, USA, 2008. ACM.

[4] Sadet Alcic and Stefan Conrad. Page segmentation by web content clustering. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 24:1–24:9, New York, NY, USA, 2011. ACM.

[5] Arvind Arasu and Hector Garcia-Molina. Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 337–348, New York, NY, USA, 2003. ACM.

[6] C. Asakawa and H. Takagi. Annotation-based transcoding for nonvisual web access. In *ASSETS'00*, pages 172–179. ACM Press, 2000.

[7] Shumeet Baluja. Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 33–42, New York, NY, USA, 2006. ACM.

[8] Ziv Bar-Yossef and Sridhar Rajagopalan. Template detection via data mining and its applications. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 580–591, New York, NY, USA, 2002. ACM.

[9] Yevgen Borodin, Jalal Mahmud, I. V. Ramakrishnan, and Amanda Stent. The hearsay non-visual web browser. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, W4A '07, pages 128–129, New York, NY, USA, 2007. ACM.

[10] Thomas Breuel. Information extraction from html documents by structural matching. In *WDA2003: Second International Workshop on Web Document Analysis*, 2003.

[11] R. Burget. Layout based information extraction from html documents. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 624 –628, sept. 2007.

[12] Radek Burget. Automatic document structure detection for data integration. In *Proceedings of the 10th international conference on Business information systems*, BIS'07, pages 391–397, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] Radek Burget and Ivana Rudolfova. Web page element classification based visual features. In *2009 First Asian conference on Intelligent Information and Database Systems*. IEEE Computer Society, 2009.

[14] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of www image search results using visual, textual and link information. In

Proceedings of the 12th annual ACM international conference on Multimedia, MULTIMEDIA '04, pages 952–959, New York, NY, USA, 2004. ACM.

[15] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-level link analysis. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 440–447, New York, NY, USA, 2004. ACM.

[16] Deng Cai, Xiaofei He, Wei ying Ma, Ji rong Wen, and Hongjiang Zhang. Organizing www images based on the analysis of page layout and web link structure. In *in The 2004 IEEE International Conference on Multimedia and EXPO*, pages 27–30. IEEE, 2004.

[17] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting content structure for web pages based on visual representation. In *Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications*, APWeb'03, pages 406–417, Berlin, Heidelberg, 2003. Springer-Verlag.

[18] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Vips: a vision based page segmentation algorithm. Technical Report MSR-TR-2003-79, Microsoft Research, 2003.

[19] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Block-based web search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 456–463, New York, NY, USA, 2004. ACM.

[20] Jiuxin Cao, Bo Mao, and Junzhou Luo. A segmentation method for web page analysis using shrinking and dividing. *International Journal of Parallel, Emergent and Distributed Systems*, 2010.

[21] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. A graph-theoretic approach to webpage segmentation. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 377–386, New York, NY, USA, 2008. ACM.

[22] J. Challenger, A. Iyengar, K. Witting, C. Ferstat, and P. Reed. A publishing system for efficiently creating dynamic web content. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 844 –853 vol.2, 2000.

[23] Nathanael Chambers, James Allen, Lucian Galescu, Hyuckchul Jung, and William Taysom. Using semantics to identify web objects. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1259–1264. AAAI Press, 2006.

[24] J. Chen, B. Zhou, J. Shi, H. Zhang, and Q. Wu. Function-based object towards website adaptation. In *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, 2001. ACM.

[25] Y. Chen, W.Y. Ma, and H.J. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[26] Yu Chen, Xing Xie, Wei-Ying Ma, and Hong-Jiang Zhang. Adapting web pages for small-screen devices. *IEEE Internet Computing*, 9:50–56, January 2005.

[27] Wen-Huang Cheng and David Gotz. Context-based page unit recommendation for web-based sensemaking tasks. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, pages 107–116, New York, NY, USA, 2009. ACM.

[28] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Very Large Data Bases*, pages 109–118, 2001.

[29] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y.-K. Ng, and R. D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data Knowl. Eng.*, 31:227–251, November 1999.

[30] D. W. Embley, Y. Jiang, and Y.-K. Ng. Record-boundary discovery in web documents. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, SIGMOD '99, pages 467–478, New York, NY, USA, 1999. ACM.

[31] David W. Embley and Li Xu. Record location and reconfiguration in unstructured multiple-record web documents. In *Selected papers from the Third International Workshop WebDB 2000 on The World Wide Web and Databases*, pages 256–274, London, UK, 2001. Springer-Verlag.

[32] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 100–110, New York, NY, USA, 2004. ACM.

[33] Fariza Fauzi, Jer-Lang Hong, and Mohammed Belkhatir. Webpage segmentation for extracting images and their surronding contextual information. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 649–652, 2009.

[34] Xiao-Dong Gu, Jinlin Chen, Wei-Ying Ma, and Guo-Liang Chen. Visual based content understanding towards web adaptation. In *AH '02: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 164–173, London, UK, 2002. Springer-Verlag.

[35] H. Guo, J. Mahmud, Y. Boroding, A. Stent, and I. V. Ramakrishnan. A general approach for partitioning web page content based on geometric and style information. In *In Proc. of ICDAR 2007*, pages 929–933, 2007.

[36] Hui Guo and A Stent. Taxonomy based data extraction from multi-item web pages. In *In Proceedings of the Workshop on Web Content Mining with Human Language Technologies at ISWC*, 2006.

[37] Aditya Gupta, Anuj Kumar, Mayank, V. N. Tripathi, and S. Tapaswi. Mobile web: web manipulation for small displays using multi-level hierarchy page segmentation.

In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, Mobility '07, pages 599–606, New York, NY, USA, 2007. ACM.

[38] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semistructured information from the web. Technical Report 1997-38, Stanford InfoLab, 1997.

[39] Gen Hattori, Keiichiro Hoashi, Kazunori Matsumoto, and Fumiaki Sugaya. Robust web page segmentation for mobile terminal using content-distances and page layout information. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 361–370, New York, NY, USA, 2007. ACM Press.

[40] X. He, J.-R. Wen D. Cai, W.-Y. Ma, and H.-J. Zhang. Imageseer: Clustering and searching www images using link and page layout analysis. Technical Report MSR-TR-2004-38, Microsoft Technical Report, 2004.

[41] Guohua Hu and Qingshan Zhao. Study to eliminating nosiy information in web pages based on data mining. In *2010 Sixth International Conferenece on Natural Computation (ICNC 2010)*, 2010.

[42] Yonghyun Hwang, Jihong Kim, and Eunkyong Seo. Structure-aware web transcoding for mobile devices. *IEEE Internet Computing*, 7(5):14–21, 2003.

[43] Christian Kohlschütter and Wolfgang Nejdl. A densitometric approach to web page segmentation. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1173–1182, New York, NY, USA, 2008. ACM.

[44] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, Marco Maggini, and Veljko Milutinovic. Recognition of common areas in a web page using visual information: a possible application in a page classification. In *Second IEEE International Conference on Data Mining (ICDM'02)*, page 250, 2002.

[45] Bernhard Krüpl and Marcus Herzog. Visually guided bottom-up table detection and segmentation in web documents. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 933–934, New York, NY, USA, 2006. ACM.

[46] Eunshil Lee, Jinbeom Kang, Joongmin Choi, and Jaeyoung Yang. Topic-specific web content adaptation to mobile devices. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 845–848, Washington, DC, USA, 2006. IEEE Computer Society.

[47] Kristina Lerman, Lise Getoor, Steven Minton, and Craig Knoblock. Using the structure of web sites for automatic segmentation of tables. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 119–130, New York, NY, USA, 2004. ACM.

[48] Shian-Hua Lin and Jan-Ming Ho. Discovering informative content blocks from web documents. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 588–593, New York, NY, USA, 2002. ACM.

[49] Bing Liu, Chee Wee Chin, and Hwee Tou Ng. Mining topic-specific concepts and definitions on the web. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 251–260, New York, NY, USA, 2003. ACM.

[50] Wenyin Liu, Xiaotie Deng, Guanglin Huang, and Anthony Y. Fu. An antiphishing strategy based on visual similarity assessment. *IEEE Internet Computing*, 10:58–65, March 2006.

[51] Yin Liu, Wenyin Liu, and Changjun Jiang. User interest detection on web pages for building personalized information agent. In Qing Li, Guoren Wang, and Ling Feng, editors, *Advances in Web-Age Information Management*, volume 3129 of *Lecture Notes in Computer Science*, pages 280–290. Springer Berlin / Heidelberg, 2004.

[52] Jalal Mahmud, Yevgen Borodin, Dipanjan Das, and I. V. Ramakrishnan. Combating information overload in non-visual web access using context. In *Proceedings of the 12th international conference on Intelligent user interfaces*, IUI '07, pages 341–344, New York, NY, USA, 2007. ACM.

[53] Jalal U. Mahmud, Yevgen Borodin, and I. V. Ramakrishnan. Csurf: a context-driven non-visual web-browser. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 31–40, New York, NY, USA, 2007. ACM.

[54] Natasa Milic-Frayling and Ralph Sommerer. Smartview: Flexible viewing of web page contents. In *Poster Proceedings of the Eleventh International World Wide Web Conference*, May 2002.

[55] Saikat Mukherjee, Guizhen Yang, and I. V. Ramakrishnan. Automatic annotation of content-rich html documents: Structural and semantic analysis. In *In Intl. Semantic Web Conf. (ISWC*, pages 533–549, 2003.

[56] Tomoyuki Nanno, Suguru Saito, and Manabu Okumura. Structuring web pages based on repetation of elements. *Transactions of Information Processing Society of Japan*, 45(9):2157–2167, 2004.

[57] Giuseppe Della Penna, Daniele Magazzeni, and Sergio Orefice. Visual extraction of information from web pages. *Journal of Visual Languages and Computing*, 21:23–32, 2009.

[58] Lakshmish Ramaswamy, Arun Iyengar, Ling Liu, and Fred Douglis. Automatic detection of fragments in dynamically generated web pages. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 443–454, New York, NY, USA, 2004. ACM.

[59] Lakshmish Ramaswamy, Arun Iyengar, Ling Liu, and Fred Douglis. Automatic fragment detection in dynamic web pages and its impact on caching. *IEEE Transactions on Knowledge and Data Engineering*, 17:859–874, 2005.

[60] Myriam Ben Saad and Stéphane Gançarski. Using visual pages analysis for optimizing web archiving. In *Proceedings of the 2010 EDBT/ICDT Workshops*, EDBT '10, pages 43:1–43:7, New York, NY, USA, 2010. ACM.

[61] Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. Learning block importance models for web pages. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 203–211, New York, NY, USA, 2004. ACM.

[62] Alex Spengler and Patrick Gallinari. Document structure meets page layout: loopy random fields for web news content extraction. In *Proceedings of the 10th ACM symposium on Document engineering*, DocEng '10, pages 151–160, New York, NY, USA, 2010. ACM.

[63] H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda. Site-wide annotation: Reconstructing existing pages to be accessible. In *ASSETS'02*, pages 81–88. ACM Press, 2002.

[64] Gujjar Vineel. Web page dom node characterization and its application to page segmentation. In *Proceedings of the 3rd IEEE international conference on Internet multimedia services architecture and applications*, IMSAA'09, pages 325–330, Piscataway, NJ, USA, 2009. IEEE Press.

[65] Xin-Jing Wang, Wei-Ying Ma, Gui-Rong Xue, and Xing Li. Multi-model similarity propagation and its application for web image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 944–951, New York, NY, USA, 2004. ACM.

[66] Liu Wenyin, Guanglin Huang, Liu Xiaoyue, Xiaotie Deng, and Zhang Min. Phishing webpage detection. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, pages 560–564, Washington, DC, USA, 2005. IEEE Computer Society.

[67] Liu Wenyin, Guanglin Huang, Liu Xiaoyue, Zhang Min, and Xiaotie Deng. Detection of phishing webpages based on visual similarity. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 1060–1061, New York, NY, USA, 2005. ACM.

[68] Y. Whang, C. Jung, J. Kim, and S. Chung. Webalchemist: A web transcoding system for mobile web access in handheld devices. In *Optoelectronic and Wireless Data Management, Processing, Storage, and Retrieval*, pages 102–109, 2001.

[69] Ou Wu, Yunfei Chen, Bing Li, and Weiming Hu. Evaluating the visual quality of web pages using a computational aesthetic approach. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 337–346, New York, NY, USA, 2011. ACM.

[70] Peifeng Xiang and Yuanchun Shi. Recovering semantic relations from web pages based on visual cues. In *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, pages 342–344, New York, NY, USA, 2006. ACM.

[71] Peifeng Xiang, Xin Yang, and Yuanchun Shi. Web page segmentation based on gestalt theory. In *Multimedia and Expo 2007 IEEE International Conference (ICME)*, 2007.

[72] Xiangye Xiao, Qiong Luo, Dan Hong, and Hongbo Fu. Slicing*-tree based web page transformation for small displays. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 303–304, New York, NY, USA, 2005. ACM.

[73] Yunpeng Xiao, Yang Tao, and Qian Li. Web page adaptation for mobile device. In *Wireless Communications, Networking and Mobile Computing*, 2008.

[74] Yunpeng Xiao, Yang Tao, and Wenji Li. A dynamic web page adaptation for mobile device based on web2.0. In *Proceedings of the 2008 Advanced Software Engineering and Its Applications*, pages 119–122, Washington, DC, USA, 2008. IEEE Computer Society.

[75] Xing Xie, Gengxin Miao, Ruihua Song, Ji-Rong Wen, and Wei-Ying Ma. Efficient browsing of web search results on mobile devices based on block importance model. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 17–26, Washington, DC, USA, 2005. IEEE Computer Society.

[76] Xin Yang and Yuanchun Shi. Enhanced gestalt theory guided web page segmentation for mobile browsing. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '09, pages 46–49, Washington, DC, USA, 2009. IEEE Computer Society.

[77] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 296–305, New York, NY, USA, 2003. ACM.

[78] X. Yin and W.S. Lee. Using link analysis to improve layout on mobile devices. In *Proceedings of the Thirteenth International World Wide Web Conference*, pages 338–344, 2004.

[79] Xinyi Yin and Wee Sun Lee. Understanding the function of web elements for mobile content delivery using random walk models. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 1150–1151, New York, NY, USA, 2005. ACM.

[80] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 11–18, New York, NY, USA, 2003. ACM.