Main Content Detection in HTML Journal Articles

Alastair R. Rae
National Library of Medicine
Bethesda, Maryland
alastair.rae@nih.gov

Jongwoo Kim
National Library of Medicine
Bethesda, Maryland
jongkim@nih.gov

Daniel Le
National Library of Medicine
Bethesda, Maryland
danle@nih.gov

George R. Thoma
National Library of Medicine
Bethesda, Maryland
gthoma@nih.gov

ABSTRACT

Web content extraction algorithms have been shown to improve the performance of web content analysis tasks. This is because noisy web page content, such as advertisements and navigation links, can significantly degrade performance. This paper presents a novel and effective layout analysis algorithm for main content detection in HTML journal articles. The algorithm first segments a web page based on rendered line breaks, then based on its column structure, and finally identifies the column that contains the most paragraph text. On a test set of 359 manually labeled HTML journal articles, the proposed layout analysis algorithm was found to significantly outperform an alternative semantic markup algorithm based on HTML 5 semantic tags. The precision, recall, and F-score of the layout analysis algorithm were measured to be 0.96, 0.99, and 0.98 respectively.

CCS CONCEPTS

• **Information systems → Data extraction and integration**;

KEYWORDS

web page segmentation, web content extraction, HTML 5

ACM Reference Format:
Alastair R. Rae, Jongwoo Kim, Daniel Le, and George R. Thoma. 2018. Main Content Detection in HTML Journal Articles. In *DocEng '18: ACM Symposium on Document Engineering 2018, August 28–31, 2018, Halifax, NS, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209280.3229115

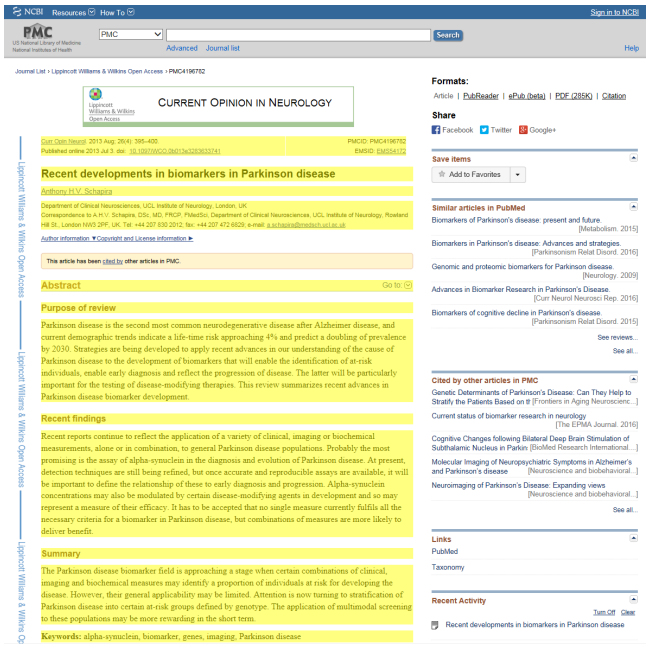

Figure 1: An HTML journal article with main content highlighted in yellow.

1 INTRODUCTION

Detecting and extracting the main content of a web page is an important pre-processing step for many web content analysis tasks. This is because noisy web page content, such as advertisements and navigation links, can significantly degrade performance.

The United States National Library of Medicine has an interest in web content analysis due to the maintenance of MEDLINE®, the preeminent bibliographic database of biomedical journal literature. With increasing numbers of journal articles published online in HTML format, it is important for the library to have automated techniques to extract and validate bibliographic data, such as grant numbers, databank accession numbers, etc.

In this paper we present a novel layout analysis algorithm for main content detection in HTML journal articles. As shown in Fig. 1, we consider the main content of an HTML journal article to be any text that is unique to the article, excluding text from HTML templates and automatically generated content.

The paper is organized as follows; Section 2 provides an overview of related work, Section 3 describes the proposed layout analysis algorithm, Section 4 compares the performance of the layout analysis algorithm to an alternative semantic markup algorithm, and Section 5 contains conclusions.

2 RELATED WORK

There are three main approaches to web page segmentation: text-based approaches, DOM-based approaches, and visual approaches.

DocEng '18, August 28–31, 2018, Halifax, NS, Canada
2018. ACM ISBN 978-1-4503-5769-2/18/08.
https://doi.org/10.1145/3209280.3229115

Text-based approaches [4] analyze the web page HTML without taking its tree structure into account, simply treating the HTML as a sequence of text and HTML tags. Text-based approaches are the fastest of the three approaches as they do not require a web page to be rendered, or the HTML to be parsed, but the disadvantage is that important structural and visual information is ignored.

DOM-based approaches [1] use an HTML parser to create a Document Object Model (DOM) and they focus on the HTML structure of a web page rather than its visual appearance. DOM-based approaches are relatively fast as they do not require a web page to be rendered, but like text-based approaches, they have the disadvantage of ignoring important style and layout information.

Visual approaches [2] analyze the style and layout of the fully rendered web page and can therefore use the same visual cues as humans for segmentation. The main disadvantage of visual approaches is that rendering a web page is slow and computationally expensive.

The three most common approaches to main content detection are heuristic approaches, machine learning approaches, and template-based approaches. Heuristic approaches [5] use manually programmed heuristic rules to identify the main content, machine learning approaches [3] learn how to detect the main content from labeled training data, and template-based approaches [1] detect the main content indirectly by identifying content that is duplicated between web pages.

A recent study [6] has shown that many existing algorithms have become somewhat obsolete due to the rapid changes in web technologies over the last 15 years. The most problematic technological trend is that an increasing number of web pages are generating their main content dynamically using JavaScript. This means that the downloaded HTML may not contain all of the web page content and in the worst case it will simply provide an entry point for the web application. This change is likely to significantly decrease the performance of any algorithm that does not render web pages in a browser. Furthermore, the study highlights the increasing adoption of semantic HTML markup and predicts that such technologies may eventually make the current generation of main content detection algorithms obsolete. They show that a simple main content detection algorithm based on the HTML 5 <article> tag has comparable performance to many published algorithms.

The algorithm proposed in this paper uses a visual approach for web page segmentation and a heuristic approach for main content detection. Its input is the fully rendered web page and therefore, unlike many previously published algorithms, it is able to process dynamically generated web page content. Other selling points of the algorithm are that it is relatively simple and it is based on a small number of fundamental HTML, Cascading Style Sheets (CSS), and text features; we therefore expect it to be robust to future changes in web technologies.

3 PROPOSED LAYOUT ANALYSIS ALGORITHM

This section describes in detail the proposed layout analysis algorithm for main content detection. The algorithm can be considered to have three high level steps; line break segmentation, column segmentation, and main content labeling. These three steps are described in subsections 3.1 - 3.3 below. The algorithm code and datasets are available online at https://github.com/raear/html-zoning.

3.1 Step 1: Line Break Segmentation

The first step of the layout analysis algorithm is an initial segmentation into zones based on the presence of line breaks in the rendered web page. The segmentation algorithm is an improved version of our previously published HTML zoning algorithm [7]. The algorithm takes inspiration from established techniques for printed document segmentation, but the approach is necessarily different because the input is the web page DOM.

The line break segmentation algorithm uses the W3C standards for HTML rendering to determine the positions of line breaks in a rendered web page. The relevant W3C standard is the CSS Visual Formatting Model which states that, within normal flow, an HTML element participates in either a block or inline formatting context. Elements that participate in a block formatting context expand to fill all available horizontal space and introduce line breaks before and after their content, while elements that participate in an inline formatting context only occupy the space that they need and do not introduce line breaks.

The formatting context for any element can be determined from the value of its CSS display property; elements with a display value of block participate in the block formatting context (*block elements*) and elements with a display value of inline participate in the inline formatting context (*inline elements*). The
 and <hr> elements (*line break elements*) are a special case because they do not contain content and they introduce line breaks directly.

The algorithm begins by rendering a web page in Internet Explorer in order to generate the DOM. The DOM is then transformed into a new structural model of the web page, which we call the *zone tree*. The zone tree is a hierarchical representation of the layout of a web page and it is composed of zones; each representing the aggregate of one or more DOM nodes. Leaf zones correspond to rectangular regions of the web page that do not contain line breaks.

The zone tree generation algorithm is a recursive algorithm that only considers visible HTML elements with text. The input to the algorithm is a zone for the <body> element. This zone is broken down into children zones based on the children of the <body> element; block elements are placed in their own *block zone* and inline elements are merged into a single *inline zone*. When a line break element is encountered, a new zone is created, and this often splits consecutive inline elements into two inline zones. The algorithm continues until the leaf zones no longer contain line breaks and this is the case when none of the zone's elements, or their descendants, introduce line breaks. An element will introduce a line break in three circumstances: when it has two or more block children, when it has one or more block children and one or more inline children, or when it has two or more inline children separated by one or more line break children.

Fig. 2 shows a typical zoning result for an HTML journal article. The figure shows that the algorithm successfully segments a web page into semantically coherent regions including the title, the author, the affiliation, section headings, and paragraphs.

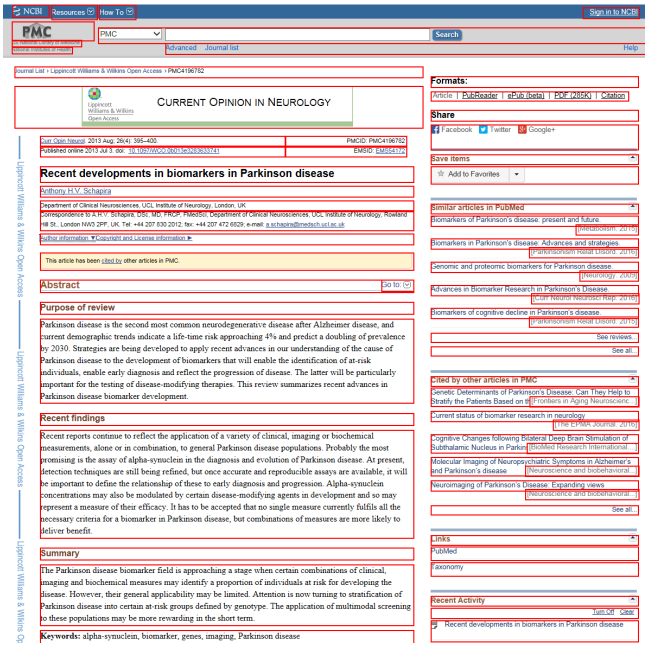

Figure 2: Leaf zones (outlined in red) for an HTML journal article.

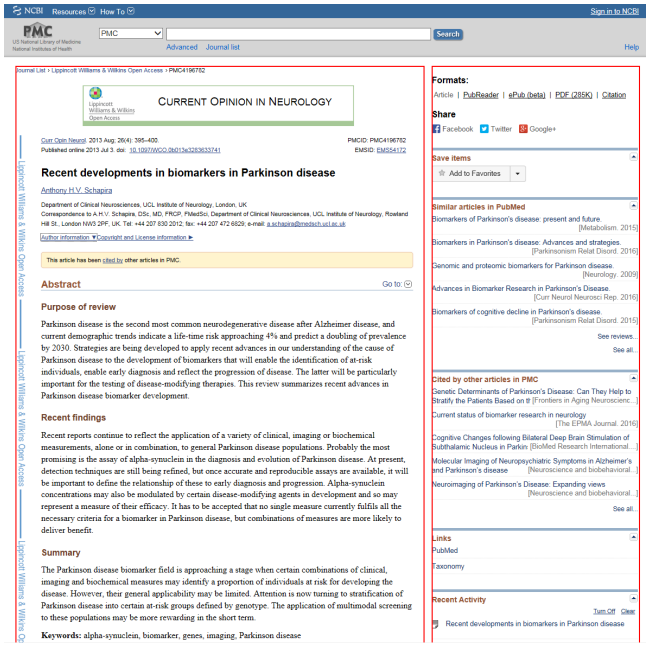

Figure 3: Detected main content column (left) and aside column (right). Columns are outlined in red.

3.2 Step 2: Column Segmentation

The input to the column segmentation algorithm is the zone tree and the output is a new higher level structural model of the web page that we call the *column tree*. The column tree represents the hierarchical column structure of the web page and it is composed of tree nodes that we call columns. A *column* is the aggregate of one or more consecutive zones that share the same left and right alignments. It corresponds to a rectangular region of the web page, characterized by its left and right coordinates.

The column tree generation algorithm is analogous to the zone tree generation algorithm; however, instead of merging inline elements into zones, zones are merged into columns if their left and right coordinates are equal within a tolerance. The tolerance is set to 10% of the column width. After the column tree has been generated, it is simplified by collapsing branches for which the parent and child columns are equal within a tolerance. To collapse a branch, the child column is replaced with its children. Fig. 3 shows the detected main content column and aside column for an HTML journal article.

3.3 Step 3: Main Content Labeling

The final step of the layout analysis algorithm is to label the column that contains the main content. Our approach is based on the observation that the main content is distinctive, in that it contains multiple paragraphs, each containing multiple sentences. The main content labeling algorithm has three high level steps and these are: paragraph labeling, main content score computation, and main content column search.

The paragraph labeling step identifies zones in the zone tree that correspond to paragraphs. Leaf zones are labeled as paragraphs if they contain two or more sentences, where a sentence is required to contain at least one verb.

The main content score computation step assigns a numerical score to each column that is proportional to the amount of main content contained by the column. We call this score the *main content score* and it is calculated based on the paragraph labels assigned in the previous step. Returning to the zone tree, leaf zones that have been labeled as paragraphs are assigned a main content score that is equal to their word count. Non-paragraph leaf zones are assigned a main content score of zero and non-leaf zones are assigned a score equal to the sum of the scores of their children. By this definition, the score of the root (<body> element) zone is equal to the total main content score and we divide all zone scores by this value to normalize them between zero and one. For the column tree, the main content score for each column is computed as the sum of the scores for the zones contained by the column. The normalized main content score of a column therefore represents the fraction of the total main content associated with the column.

In the main content column search step, the computed main content score is used to search for the column in the column tree that contains the main content. An analysis of the column trees generated for many different HTML journal articles showed that the main content column is very often associated with a sudden drop in main content score as the tree is traversed from the main content column to its children. This is because the main content column is associated with most of the detected main content, whereas its children typically represent a fragmented column substructure, with each child associated with much less of the total main content.

Our approach to identify the main content column is therefore as follows. The column tree is traversed using a top-down strategy that

Table 1: Comparison of the main content detection performance of the layout analysis algorithm and the semantic markup algorithm. The "ST Only" test set is the subset of articles containing either the <article> or <main> tags.

Algorithm	Test Set	Precision	Recall	F-score
Layout analysis	Full	0.96	0.99	0.98
Semantic markup	Full	0.43	0.43	0.42
Layout analysis	ST only	0.97	0.99	0.98
Semantic markup	ST only	0.97	0.97	0.95

follows the path with the highest main content score. Specifically, each traversal step is between the current node and the child node with the highest main content score. For each step, a metric called the *main content score delta* is computed and this is defined as the difference between the main content score for the current node and the main content score for the next node. If this delta value is greater than an empirically determined threshold of 0.3, then the current column and all of its associated zones are labeled as main content.

4 EVALUATION

In this section we evaluate the performance of the layout analysis algorithm and compare its performance to an alternative semantic markup algorithm.

The semantic markup algorithm detects the main content using HTML 5 semantic tags and it is similar to the semantic tag algorithm proposed in [6]. Two HTML 5 semantic tags are of particular interest for main content detection: the <article> tag and the <main> tag. The <article> tag defines independent and self-contained content while the <main> tag defines the main content that is unique to the web page. For use as a baseline, we developed a configurable algorithm that can use either tag name for main content detection. The algorithm was implemented as follows. First, a line break segmentation is performed (Section 3.1). Next, the resulting zone tree is searched, using a top-down breadth-first traversal, to find the first zone that contains an element with the semantic tag name of interest. If such a zone is found, the zone and its descendants are labeled as main content. With the increasing adoption of HTML 5, the semantic markup algorithm is a practical alternative to more complex main content detection algorithms and the approach has been observed to have comparable performance to previously published algorithms [6].

The performance evaluation was conducted using a test set of 359 manually labeled HTML journal articles from 120 different journals. The test set contains a wide variety of different web page formats and is representative of the web page formats of journals in MEDLINE. For each article, a ground truth was created by manually labeling leaf zones as main content. The main content detection performance was evaluated using the standard performance metrics of precision, recall, and F1-score. These metrics were computed based on the number of correctly labeled words, where each word takes the label of its zone. All presented performance metric values are the mean values for the test set.

Table 1 compares the main content detection performance of the layout analysis algorithm and the semantic markup algorithm. Performance measures are presented for both the full test set of 359 articles and the subset of 158 articles that contained either the <article> or <main> semantic tags. For articles containing both semantic tags, we chose the semantic markup algorithm configuration (<article> or <main> tag) that gave the highest F-score. The table shows that the layout analysis algorithm significantly outperforms the semantic markup algorithm on the full test set and this is mainly because the HTML 5 semantic tags of interest were only present in 44% of articles. On the subset of articles that contained either the <article> or <main> tags the performance of the two algorithms is more similar but the layout analysis algorithm still has a higher F-score of 0.98, compared to 0.95. The main reason for the lower performance of the semantic markup algorithm is that the <article> tag is sometimes misused by web page authors.

5 CONCLUSIONS

This paper has presented a novel layout analysis algorithm for main content detection in HTML journal articles. The performance of the algorithm was evaluated using a test set of 359 manually labeled HTML journal articles and the precision, recall, and F-score were measured to be 0.96, 0.99, and 0.98 respectively. The algorithm was also found to significantly outperform an alternative semantic markup algorithm based on HTML 5 semantic tags. As the adoption of HTML 5 increases, the performance of the semantic markup algorithm will likely improve, but as highlighted by this study, the performance of such algorithms is dependent on the correct use of semantic markup technology by web page authors.

ACKNOWLEDGMENTS

This research was supported by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and Lister Hill National Center for Biomedical Communications (LHNCBC).

REFERENCES

[1] Ziv Bar-Yossef and Sridhar Rajagopalan. 2002. Template Detection via Data Mining and Its Applications. In *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*. ACM, New York, NY, USA, 580–591. https://doi.org/10.1145/511446.511522

[2] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. *VIPS: A Vision based Page Segmentation Algorithm.* Microsoft Research Technical Report MSR-TR-2003-79. Microsoft Research Asia, Beijing, China.

[3] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate Detection Using Shallow Text Features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM '10)*. ACM, New York, NY, USA, 441–450. https://doi.org/10.1145/1718487.1718542

[4] Christian Kohlschütter and Wolfgang Nejdl. 2008. A Densitometric Approach to Web Page Segmentation. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. ACM, New York, NY, USA, 1173–1182. https://doi.org/10.1145/1458082.1458237

[5] Fei Sun, Dandan Song, and Lejian Liao. 2011. DOM Based Content Extraction via Text Density. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*. ACM, New York, NY, USA, 245–254. https://doi.org/10.1145/2009916.2009952

[6] Tim Weninger, Rodrigo Palacios, Valter Crescenzi, Thomas Gottron, and Paolo Merialdo. 2016. Web Content Extraction: A MetaAnalysis of Its Past and Thoughts on Its Future. *SIGKDD Explor. Newsl.* 17, 2 (Feb. 2016), 17–23. https://doi.org/10.1145/2897350.2897353

[7] Jie Zou, Daniel Le, and George R. Thoma. 2010. Locating and Parsing Bibliographic References in HTML Medical Articles. *Int. J. Doc. Anal. Recognit.* 13, 2 (June 2010), 107–119. https://doi.org/10.1007/s10032-009-0105-9