Web Page Segmentation Revisited: Evaluation Framework and Dataset

Johannes Kiesel
johannes.kiesel@uni-weimar.de
Bauhaus-Universität Weimar

Florian Kneist
fkneist@gmail.com
Bauhaus-Universität Weimar

Lars Meyer
lars.meyer@uni-weimar.de
Bauhaus-Universität Weimar

Kristof Komlossy
kristof.komlossy@kritten.org
Bauhaus-Universität Weimar

Benno Stein
benno.stein@uni-weimar.de
Bauhaus-Universität Weimar

Martin Potthast
martin.potthast@uni-leipzig.de
Leipzig University

ABSTRACT

Each web page can be segmented into semantically coherent units that fulfill specific purposes. Though the task of automatic web page segmentation was introduced two decades ago, along with several applications in web content analysis, its foundations are still lacking. Specifically, the developed evaluation methods and datasets presume a certain downstream task, which led to a variety of incompatible datasets and evaluation methods. To address this shortcoming, we contribute two resources: (1) An evaluation framework which can be adjusted to downstream tasks by measuring the segmentation similarity regarding visual, structural, and textual elements, and which includes measures for annotator agreement, segmentation quality, and an algorithm for segmentation fusion. (2) The Webis-WebSeg-20 dataset, comprising 42,450 crowdsourced segmentations for 8,490 web pages, outranging existing sources by an order of magnitude. Our results help to better understand the "mental segmentation model" of human annotators: Among other things we find that annotators mostly agree on segmentations for all kinds of web page elements (visual, structural, and textual). Disagreement exists mostly regarding the right level of granularity, indicating a general agreement on the visual structure of web pages.

ACM Reference Format:
Johannes Kiesel, Florian Kneist, Lars Meyer, Kristof Komlossy, Benno Stein, and Martin Potthast. 2020. Web Page Segmentation Revisited: Evaluation Framework and Dataset. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412782

1 INTRODUCTION

Web page *layout* is the arrangement of content elements as semantic units which appear sensible to the human observer; web page *segmentation* is the inverse operation and means to identify these semantic units. Automatic web page segmentation is straightforward if all semantic units are prescribed explicitly in the source code. In practice, however, semantic coherence can often be inferred from the layout only, and leading approaches to web page segmentation thus employ visual features to reach an acceptable performance.

Web page segmentation has been applied for various purposes throughout the information retrieval pipeline and beyond: Our review of related work in Section 2—as well as the reviews of Fernandes et al. [16], Akpinar and Yesilada [1], and Bing et al. [5]—show that web page segmentation is used to improve crawling (template, duplicate, and change detection), information extraction (indexing, snippet generation, summarization, main content extraction, entity mining), page analysis (link analysis, design mining), and page synthesis (mobile screen adaptation, screen reading).

But despite the many publications that employ web page segmentation, the segmentation approaches have hardly been evaluated. Rather, the approaches have been judged "implicitly" by the increased performance induced in some downstream task that employs segmentation. Similarly, most segmentation algorithms have not been compared directly, and, in particular, no recent evaluations are at hand despite the constant evolution of web layouts. One reason for the missing evaluation is the lack of standard performance metrics for web page segmentation as well as suitable datasets in terms of size, diversity, and completeness of resources.

With this paper we lay new foundations for the large-scale evaluation of web page segmentation algorithms. Our main contributions are as follows: (1) We present an evaluation framework for web page segmentation that builds upon a single similarity measure for segmentations. This measure is applicable for the calculation of annotator agreement, the fusion of segmentations into a ground truth, and the evaluation of a segmentation against such a ground truth. The framework can be adapted to specific downstream tasks and is publicly available.[1] (2) Based on a reference dataset of 8,490 archived web pages we construct the Webis Web Segmentation Corpus 2020 (Webis-WebSeg-20), a publicly available dataset of 42,450 manually created web page segmentations (five per page), via crowdsourcing.[2] This dataset outranges prior resources by an order of magnitude while being more objective at the same time through the use of five independent segmentations.

In what follows, Section 2 reviews the related work. Section 3 introduces the concept of web page segments. Section 4 develops our framework for web page segmentation from this concept. Building upon these foundations, Section 5 presents our dataset including its crowdsourcing, derivation of ground truth, and analysis.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412782

[1] Code: https://github.com/webis-de/CIKM-20

[2] Data: https://doi.org/10.5281/zenodo.3354902

2 RELATED WORK

Research on web page segmentation goes back almost two decades. First defined by Kovacevic et al. [23], similar problems have even been tackled beforehand in information extraction (e.g., cf. [15]). Still, the community has not agreed on evaluation procedures, nor created commonly used benchmarks, as we detail below.

Algorithms for Web Page Segmentation. Algorithms use structural features based on the DOM tree and the textual content, and visual features extracted from renderings of individual nodes as well as the entire web page. Most algorithms use the DOM tree structure in some way, for example to identify headings [26], block nodes [1, 6], or regularities [16], and to compute the tree depth [24] or the tree distance [18] of nodes. Other algorithms use the text density [22] or visual appearance of DOM nodes when rendered (e.g., their size or color; Baluja [4], Zeleny et al. [37]). Few algorithms exclusively exploit visual cues, e.g., using edge detection on screenshots [8, 12]. Indeed, recent publications argue that only visual features provide for the necessary robustness for a generalizable algorithm [12, 37], but this claim has not been verified. Our dataset provides the resources required by all the various approaches, enabling a fair and comprehensive comparison. Although a hierarchical segmentation of a web page is conceivable, and although some algorithms hierarchically split a web page into smaller segments (e.g., [4–6]), all proposed algorithms output a single segmentation per page. We therefore adopt this view, and leave hierarchical segmentation for future work.

Datasets for Web Page Segmentation. Several datasets have been created for web page segmentation, but none has become a standard benchmark. Instead, most algorithms come with a new dataset for their evaluation (cf. Table 1). Issues that prevent the reuse of the existing datasets include missing data sources (e.g., no screenshots), bias due to heuristic annotations, no ground truth annotations, unavailability, and a non-representative sample (e.g., only a few specific websites). None of the previously published datasets combines completeness, reliability, diversity, and scale. Even the very large dataset of Fernandes et al. [16] lacks diversity, since their annotation process presumes all web pages to be homogeneous. Regarding tools for manual segmentation, only Sanoja and Gançarski [31] have proposed one: it allows to create, resize, and move rectangles on a screenshot to specify segments. Inspired by their approach, we integrate our version with Mechanical Turk to enable manaul segmentation at scale via crowdsourcing.[3]

Evaluation of Web Page Segmentation. Previous attempts to evaluate web page segmentations fall short in some respects or others. Some resort to a posteriori human judgment of detected segments (e.g., Cai et al. [6]), which does not scale well and yields hardly replicable measurements. Others evaluate based on the web page's text only, which allows for using existing evaluation measures for this task (e.g., Kohlschütter and Nejdl [22], Manabe and Tajima [28]), but restricts the evaluation to text-only segments. Yet others measure the overlap between an automatic segmentation and a ground truth [37], or count matching cases (one-to-one, one-to-many, zero-to-one, etc.; Sanoja and Gançarski [32]). Such matching measures, however, unfairly handle cases of over- and undersegmentation: The measure proposed by Zeleny et al. [37] penalizes splitting a ground truth segment into several small ones more than returning just one of the small segments. The measure of Sanoja and Gançarski does not penalize splitting a ground truth segment at all, making it trivial to achieve the maximum score.

Table 1: Overview of existing segmentation datasets; their enclosed data: </> HTML code, 🗜 resources (CSS, images, etc.), 🖥 screenshot; segment annotations: ✎ manual, ⧉ heuristic, or a posteriori judgment; availability: 🔗 publicly, ✉ on request, or not anymore as per correspondence with the authors; and the numbers of websites (if given) and web pages.

Author	Reference	Year	Characteristics	Sites	Pages
Kovacevic et al.	[23]	2002	</> ✎	n/a	515
Cai et al.	[6]	2003	</>	n/a	140
Vadrevu et al.	[36]	2005	</>	n/a	240
Hattori et al.	[18]	2007	</> ✎	100	100
Chakrabarti et al.	[9]	2008	</> ✎	n/a	105
Kohlschütter and Nejdl	[22]	2008	</> ✎	102	111
Cao et al.	[8]	2010	</> 🗜 🖥	n/a	20
Spengler and Gallinari	[35]	2010	</> 🗜 🖥 ✎	177	604
Fernandes et al.	[16]	2011	</> ⧉	15	457,542
Pasupathi et al.	[30]	2012	</>	10	15
Sanoja and Gançarski	[31]	2013	</> 🗜 🖥 ✎ ✉	n/a	100
Bing et al.	[5]	2014	</> ✎	n/a	1,000
Kreuzer et al.	[24]	2015	</> 🗜 ✎ 🔗	59	152
Manabe and Tajima	[28]	2015	</> 🗜 ✎ 🔗	981	1,219
Sanoja and Gançarski	[32]	2015	</> ✎ 🔗	125	125
Cormier et al.	[13]	2016	🖥 ✉	50	50
Cormier et al.	[12]	2017	🖥 ✉	100	100
Sanoja and Gançarski	[33]	2017	</> 🗜 🖥 ✎ ✉	n/a	40
Zeleny et al.	[37]	2017	</> 🗜 🖥 ⧉	5	800
Andrew et al.	[3]	2019	</> 🗜 ✎ ✉	n/a	50
Webis-WebSeg-20		2020	</> 🗜 🖥 ✎ 🔗	4,824	8,490

While most authors use evaluation measures of some kind to assess how well an automatic segmentation matches a human one, the assessment of human segmentations is typically lacking. For most datasets, each page was annotated by a single annotator only. While Zeleny et al. [37] employed three annotators per page, they treat each segmentation as alternative ground truth. Manabe and Tajima [28] calculated the annotator agreement for a few test pages, but only for segments that directly correspond to HTML block elements. An algorithm to fuse annotations of different workers into a single ground truth segmentation has not been considered.

Segmentation Outside the Web. Beyond web pages, segmentation tasks are studied for scanned print documents and generic images. Unlike for web pages, typically no semi-structured representation like the HTML source is available for either. At ICDAR, a long-running competition addresses the segmentation of scanned print documents featuring complex typesetting [11]. However, there is much less ambiguity about the level of granularity in print documents and the evaluation measures thus focus on segment matching. Generic image segmentation (e.g., of photos) is often cast as an object recognition task: images rarely contain text and objects are rarely rectangular. Unlike for web page segmentation, the evaluation measures employed for generic image segmentation match boundary pixels [29] or objects directly, using huge datasets of hundreds of thousands of images like Microsoft COCO [27].

[3] Mechanical Turk features a built-in image segmentation interface, but we found that an interface tailored to web page segmentation allows for a much quicker annotation.

3 CONCEPT FORMATION: PAGE SEGMENT

Nine of the 19 publications listed in Table 1 give—explicitly or implicitly—a definition of what a web page segment is. The most common one (though used in only four publications) is that of a visual "block" with coherent content [9, 24, 26, 37]. Other definitions characterize segments by their edges [12, 13], as being semantically self-contained [16], as distinct [30], or as labeled with a heading [28]. Only two papers resort to HTML/DOM elements or sub-trees as segment building blocks [9, 24]. Seven of the nine definitions require a segment to be cohesive, and two define a segment as being "different" to other parts of a web page. Most of the definitions do not include information about the desired level of granularity, probably because different downstream applications have different requirements [37].

Altogether, the concept of a segment is not precisely captured: Does an individual menu item count as a segment, or need all menu items be combined, or is the entire sidebar to which the menu belongs the "true" segment? It is also unclear whether a more precise specification would be meaningful across web page genres. Note in this regard that even the terminology to describe granularity levels is used inconsistently: Kreuzer et al. [24], for instance, differentiate between high-level and sub-level segments, while other authors resort to exemplifying the desired level of granularity, such as "header", "left menu", etc., as in Kovacevic et al. [23].

In light of the ambiguities and limitations of the existing segment definitions, we refrained from proposed a tenth definition, but opted instead for a concept formation approach based on crowdsourcing. For this purpose, each page in our dataset has been annotated by five annotators, providing us with a rich source of information to analyze what a human onlooker considers a plausible segment and granularity level, respectively:

> A web page segment is a part of a web page containing those elements that belong together as per agreement among a majority of viewers.

This concept of a web page segment is grounded on well-known layout patterns ("header", "main", "footer", etc.) and human perception habits (such as Gestalt principles like proximity [17]). As Section 5.2 shows, we indeed get strong agreement among independent annotators. Our dataset thus allows for the development of web page segmentation algorithms that operationalize this concept.

4 FRAMEWORK FOR PAGE SEGMENTATIONS

The creation and usage of a dataset that adheres to the concept of page segments as introduced above requires answers to the following three questions: How to measure the agreement of users? How to fuse single segmentations into a coherent ground truth? How to evaluate this ground truth?

As shown at the end of this section, the answer to all of these questions boils down to measuring the similarity of two page segmentations. To choose a measure of segmentation similarity, we cast web page segmentation as a clustering task and draw from the theoretical foundations of cluster similarity measures. In order to identify the objects that are to be clustered into clusters corresponding to page segments, we begin by studying alternative candidates for atomic elements of a web page.

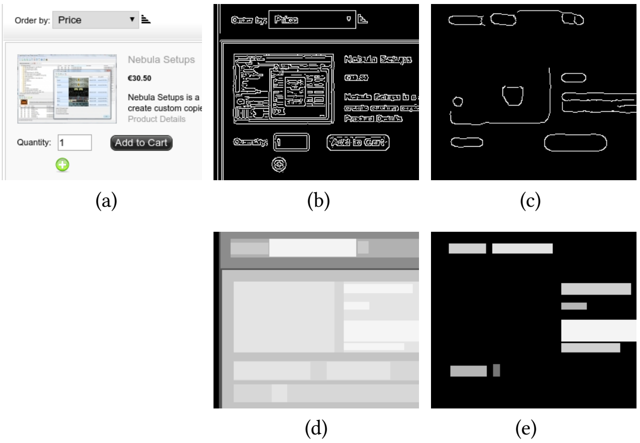


Figure 1: Visualization of atomic elements for (a) an exemplary page excerpt: (b) fine- and (c) coarse-grained edges; (d) DOM nodes; and (e) characters (text nodes). Lighter pixels indicate more elements at the respective position in (a). The image for all pixel elements would be completely white.

4.1 Atomic Elements of a Web Page

The first component of our framework is the selection of the "atomic" elements of a web page—the nature of which is deliberately left open in our concept of a web page segment. We identify three alternative sets of atomic elements that can be clustered to form segments of a web page: (1) pixels, (2) DOM nodes, and (3) characters. Besides the entire sets, also defined subsets might be considered dependent on the downstream task; for instance, background pixels may be considered unimportant. As our dataset is task-agnostic, we repeat all analyses for a selection of five sets that cover a variety of intuitions, namely three pixel subsets, and one each of DOM nodes and characters (see Figure 1 for an illustration):

- **Pixels.** The three pixel subsets include (1) all pixels of a web page's screenshot, and pixels at (2) fine-grained, and (3) coarse-grained visual edges as per Canny's edge detection algorithm [7], which is best-suited for web pages [8]. Fine-grained edges include the outline of characters at 10pt font size, and coarse-grained edges only lines of text.[4] Edges are here used as an indicator for the content density of segments.
- **DOM nodes.** All visible DOM nodes of a web page, i.e., element and text nodes. As the DOM is organized hierarchically, more nodes lie in regions that are more deeply structured.
- **Characters.** All characters on a web page.

These sets of elements capture intuitive choices for generic image-based web page segmentations (all pixels, e.g., for design mining), for specific image-based segmentation where background pixels are irrelevant (edge pixels, e.g., for mobile screen adaptation), for structure-based segmentation (visible DOM nodes, e.g., for information extraction), and for text-based segmentation (characters, e.g., for screen reading).

[4] Both parameter sets have the same radius of 0 and lower percentage of 1. For fine-grained edges, the upper percentage is 2 and $\sigma = 1$; for coarse-grained edges, the upper percentage is 16 and $\sigma = 5$ to counteract the increased roughness of edge lines.

4.2 Web Page Segmentation Similarity

Like in clustering, measuring segmentation similarity in our framework is agnostic of the nature of the atomic elements. Given a web page p, let $E = \{e_1, \ldots, e_n\}$ be the set of its atomic elements. Then $S = \{s_1, \ldots, s_m\}$ denotes a (possibly partial and/or overlapping) segmentation of p into segments $s_i \subseteq E$. Given two segmentations S and S^* of the same page p, we identify the extended BCubed F_1-score by Amigó et al. [2], F_{B^3}, an extrinsic cluster evaluation measure, as particularly suited to measure segmentation similarity. Specifically, F_{B^3} has been shown to fulfill all requirements for a segmentation similarity measure that we gathered in our literature review and some more [2]: it handles both partial segmentations and overlapping—and thus also nested—segments, and is robust to trivial segmentations (e.g., using every pixel as own segment, or one segment that covers the entire page). Moreover, as F_{B^3} is symmetrical, it can be used as a generic segmentation similarity measure, too.

Analogous to the well-known F_1-score, F_{B^3} is the harmonic mean of the extended BCubed precision (P_{B^3}) and recall (R_{B^3}):

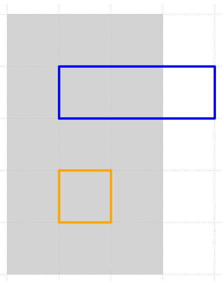
$$P_{B^3}(S, S^*) = \frac{1}{\left|E^S\right|} \sum_{e \in E^S} \left(\frac{1}{\left|E^S_e\right|} \sum_{e' \in E^S_e} \left(\frac{\min\left(|S_e \cap S_{e'}|, |S^*_e \cap S^*_{e'}|\right)}{|S_e \cap S_{e'}|} \right) \right);$$

$$R_{B^3}(S, S^*) = P_{B^3}(S^*, S); \quad F_{B^3}(S, S^*) = \frac{2 \cdot P_{B^3}(S, S^*) \cdot R_{B^3}(S, S^*)}{P_{B^3}(S, S^*) + R_{B^3}(S, S^*)};$$

where $S_e \subseteq S$ is the subset of segments that contain element e, $E^S \subseteq E$ is the subset of elements that are part of at least one segment in S, and $E^S_e \subset E$ is the subset of elements that accompany element e in at least on segment in S. Formally, $S_e = \{s \mid s \in S \wedge e \in s\}$, $E^S = \{e \mid e \in E \wedge S_e \neq \emptyset\}$, and $E^S_e = \{e' \mid e' \in E \wedge S_e \cap S_{e'} \neq \emptyset\}$. For illustration, consider the case of non-overlapping segments: $|S_e \cap S_{e'}|$ is 1 if and only if e and e' are in the same segment in S, whereas $\min(|S_e \cap S_{e'}|, |S^*_e \cap S^*_{e'}|)$ is 1 if and only if they are in the same segment in both S and S^*.

Precision (P_{B^3}) and recall (R_{B^3}) offer helpful insight into the achieved F_{B^3}. Specifically, P_{B^3} ignores errors of strict oversegmentation (i.e., a ground truth segment is split), while R_{B^3} ignores errors of strict undersegmentation (i.e., ground truth segments are merged). Therefore, by comparison with F_{B^3}, these measures show the extent of over- and undersegmentation, and can thus directly inform the parameter optimization of the employed approach.

Optimizations. The number of operations to calculate F_{B^3} grows quadratically with the number of atomic elements, so we developed optimizations to speed up the calculation. For pixels as atomic elements, we determine all largest regions where no segmentation divides these regions. The fraction in the calculation of P_{B^3} is the same for all elements of such a region. Therefore, we need to calculate this fraction only once for each pair of regions and just need to multiply the result by the product of the regions' areas. Figure 2 shows a toy example to exemplify the calculation. The same applies when using edges as atomic elements, except for using only the number of edge pixels in the regions instead of the area. For characters as atomic elements, we resort to DOM text nodes weighted by the number of characters within—analogous to how we use areas instead of pixels. Note that this method is an approximation for the very rare segmentation approaches that could potentially divide up text nodes in segmentation. Unlike in the images that an edge detector produces, visual edges actually have no width, which frequently causes the edges of DOM nodes to appear a bit outside of the nodes' area in a generated image. To account for this fact, we grow the regions by two pixels before counting the number of contained edge pixels, which we tested to indeed capture nearly all relevant edge pixels.



$$P_{B^3} = \frac{\boxed{2}}{4}\left(\frac{\boxed{2}}{3} \cdot 1 + \frac{\boxed{1}}{3} \cdot 0\right) + \frac{\boxed{1}}{4}\left(\frac{\boxed{2}}{3} \cdot 0 + \frac{\boxed{1}}{3} \cdot 0\right)$$
$$+ \frac{\boxed{1}}{4}\left(\frac{\boxed{1}}{1} \cdot 1\right) = \frac{7}{12}$$
$$R_{B^3} = \frac{12}{15}\left(\frac{12}{15} \cdot 0 + \frac{\boxed{2}}{15} \cdot 0 + \frac{\boxed{1}}{15} \cdot 0\right)$$
$$+ \frac{\boxed{2}}{15}\left(\frac{12}{15} \cdot 0 + \frac{\boxed{2}}{15} \cdot 1 + \frac{\boxed{1}}{15} \cdot 0\right)$$
$$+ \frac{\boxed{1}}{15}\left(\frac{12}{15} \cdot 0 + \frac{\boxed{2}}{15} \cdot 0 + \frac{\boxed{1}}{15} \cdot 1\right) = \frac{5}{225}$$

Figure 2: Toy example for calculating the BCubed precision and recall of a segmentation with two segments (colored frames) compared to a one-segment ground truth (shaded area) for pixels. The equations are highlighted with the corresponding color frames and shades.

4.3 Application of the Segmentation Similarity

The measure of segmentation similarity introduced above can be applied to answer the three questions from the start of this section. Our released code contains a program for each of them.

Annotator Agreement. To judge whether two or more annotators were able to work consistently, their agreement is measured. Using F_{B^3}, we follow the example of popular agreement measures for text annotations like Krippendorff's α [25] and compute the average pairwise similarity of the segmentations. Specifically, for a set of segmentations $\mathcal{S}$ of the same web page, we define segmentation agreement as follows:

$$\text{Agreement}(\mathcal{S}) = \frac{1}{|\mathcal{S}| \cdot (|\mathcal{S}| - 1)} \sum_{S \in \mathcal{S}} \sum_{S' \in \mathcal{S} \setminus S} F_{B^3}(S, S')$$

The agreement of an entire dataset is then calculated as the average agreement over all web pages.

In order to analyze how much of the quantified disagreement (as calculated using F_{B^3}) is due to different annotation granularities, F_{B^3} can be replaced by $\max(P_{B^3}, R_{B^3})$. As mentioned above, P_{B^3} and R_{B^3} ignore errors from strict over- and undersegmentation, respectively. Therefore, an additional analysis with $\max(P_{B^3}, R_{B^3})$ yields insights into whether annotators disagreed on which elements belong together, and on the level of granularity.

Segmentation Fusion. Our concept of web page segments stated above is based on the majority agreement, and our framework employs F_{B^3} to fuse several segmentations into such a majority segmentation. Fused segments should contain those atomic elements which also the majority of annotators put in one segment. This intuition provides for a similarity of two atomic elements: the fraction of annotators who put those two elements in one segment.

We then fuse elements and segments with a similarity exceeding a threshold (θ_s) of one half. This fusion process corresponds to the well-known family of hierarchical agglomerative clustering algorithms [19], which relies on the similarities only and does not need a vector representation of segments. Further following the principle of majority, we fuse just those atomic elements that are in segments for a majority of annotators. We analyze the effect of both θ_s and this annotator threshold for our dataset in Section 5.3.

The choice of a specific hierarchical clustering algorithm matters only in the rare case of disagreeing majorities.[5] In such cases, the most basic hierarchical algorithms segment all elements together (single-link), or arbitrarily choose the segmentation of one majority (complete-link). Since both is not desirable, we employ the simple average or UPGMA algorithm [34], which tends toward the segmentation of the majority that groups more elements together.

Segmentation Evaluation. Analogous to its purpose in clustering, we use F_{B^3} as a measure of the quality of some segmentation S compared to a ground truth segmentation S^* of the same page.

5 THE WEBIS-WEBSEG-20 DATASET

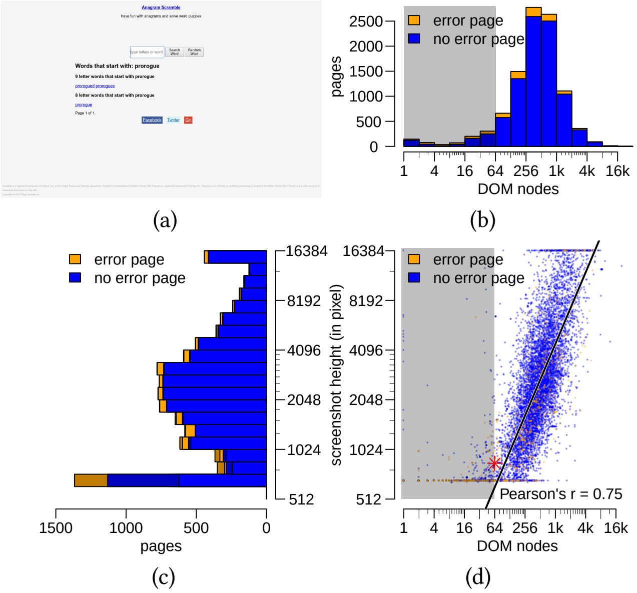
Starting point for the construction of our dataset is the Webis-Web-Archive-17 [21]. It is a web archive comprising 10,000 pages from 5,516 sites, obtained via a stratified sample from top-ranked and low-ranked sites as per their Alexa ranking (alexa.com). Our dataset has been constructed in three steps: preprocessing, human annotation, and segmentation fusion.

5.1 Preprocessing and Web Page Analysis

Although the web pages of our dataset are already contained in the original web archive, not all resources for web page segmentation are readily available, and not all pages in the archive are suited for a web page segmentation dataset. Specifically, we reproduced all pages within a browser and extracted all DOM nodes from the rendered pages, their textual content, and their locations (i.e., bounding boxes) on the accompanying screenshot. In spot checks on 100 pages, we manually verified that the locations are accurate.

During our review, we identified two problematic cases of web pages with respect to segmentation, namely simple pages and error pages. We use "simple" to refer to web pages that do not have enough content to justify a segmentation, and which we therefore exclude from our dataset. Similarly, error pages are pages which clearly miss or have wrong main content. We expect that, in a page analysis pipeline, such pages will be identified and recrawled ahead of the segmentation. We thus omitted 866 simple pages and 644 error pages, which were identified via an analysis of page complexity outlined below, and a public list of manual error annotations for the original archive [20], respectively.

To check that the dataset represents a broad sample of web pages, and to investigate page complexity, we analyze the amount of DOM nodes and the pixel height.[6] Further spot checks confirmed the intuition that simple pages have only a few DOM nodes, which allows to adjust the threshold for inclusion in our dataset accordingly: Figure 3a shows a page bordering on simplicity. In Figure 3b and 3c, we observe a seemingly natural log normal distribution both for pages across the amount of DOM nodes and pixel heights. The exceptional high number of pages with a height of 16,384 pixels is due to infinite scrolling pages, where the archiving tool stopped scrolling. Error pages follow somewhat the overall distribution of pages. As one would assume, the correlation of number of DOM nodes and pixel height (Figure 3d) is fairly strong, as indicated by the fitted log-linear model (straight line) and Pearson correlation.



Anagram Scramble

have fun with anagrams and solve word puzzles

type letters or word | Search Word | Random Word

Words that start with: prorogue

9 letter words that start with prorogue

prorogued prorogues

8 letter words that start with prorogue

prorogue

Page 1 of 1.

Scrabble is a registered trademark of Hasbro, Inc. in the United States and Canada; elsewhere, Scrabble is trademarked by Mattel. Words With Friends is a registered trademark of Zynga Inc. This site is not officially or unofficially endorsed or related to Scrabble, Words With Friends or any other product or trademark mentioned on this site.

Copyright © 2017 Page Scholar Inc.

(a) (b) (c) (d)

Figure 3: (a) Example page bordering on simplicity; (b,c) page frequency distribution, and (d) scatter plot over DOM nodes and pixel height. The page of (a) is marked as a red star in (d). The simple pages are within the shaded area and the shaded bars, respectively, and error pages are depicted orange.

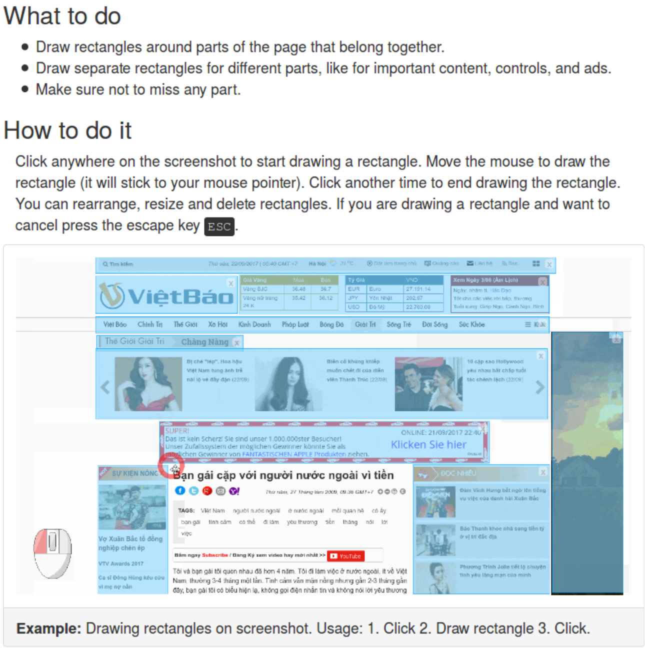
5.2 Human Annotation

For humans, segmenting a single web page is fairly straightforward. As Kreuzer et al. [24] observe: "Human beings are very good at partitioning: even if a website is in a language we are not familiar with, it is clear to us what is an advertisement, what is a menu, and so on." In order to scale up such manual segmentation to 8,490 web pages while avoiding annotator bias (e.g., systematic errors), we employ crowdsourcing. We used Amazon's Mechanical Turk and developed a tailored annotator interface that allows for drawing bounding boxes on web page screenshots, as well as a reviewer interface that allows for quality control by visualizing the segmentations. We further developed a reliable mapping of hand-drawn segments to their corresponding DOM nodes and assessed the annotation quality through measuring inter-annotator agreement.

Task Setup. Amazon's Mechanical Turk is a crowdsourcing market where requesters, like ourselves, advertise so-called "human intelligence tasks" (HITs) to workers for a per-task payment upon successful completion. In pilot experiments, we found that our task does not require expert workers, so we just required workers to have at least 100 previously approved HITs—a very low bar.

[5] Given three element sets, E_1, E_2, E_3, let $\alpha(E_i)$ be the fraction of annotators that put all elements of E_i in one segment. Then "disagreeing majorities" in its simplest form is a case where $\alpha(E_1 \cup E_2) > 0.5$, $\alpha(E_2 \cup E_3) > 0.5$, but also $\alpha(E_1 \cup E_2 \cup E_3) < 0.5$.

[6] All pages have the same width of 1366 pixels as per the web archiver employed.



What to do

- Draw rectangles around parts of the page that belong together.
- Draw separate rectangles for different parts, like for important content, controls, and ads.
- Make sure not to miss any part.

How to do it

Click anywhere on the screenshot to start drawing a rectangle. Move the mouse to draw the rectangle (it will stick to your mouse pointer). Click another time to end drawing the rectangle. You can rearrange, resize and delete rectangles. If you are drawing a rectangle and want to cancel press the escape key ESC.

Example: Drawing rectangles on screenshot. Usage: 1. Click 2. Draw rectangle 3. Click.

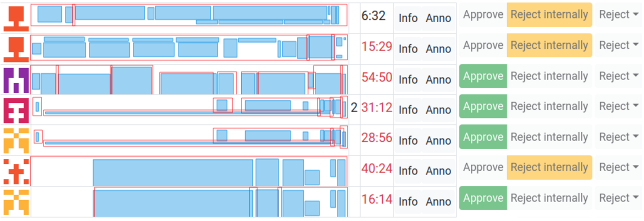
Figure 4: Annotator instructions including a GIF animation which exemplifies the task and annotation process.

To ensure equal workload per HIT despite the vastly different pixel heights of the screenshots (see Figure 3), we employed a bin-packing algorithm to distribute the web pages so that every HIT contained web pages that have a combined pixel height of approximately 16,384 pixels, the maximum pixel height of the screenshots in our dataset. On average, a HIT contained five web pages. During our pilot experiments, we determined that workers needed 11.2 minutes per HIT on average. The payment per HIT was set to \$0.75 for an hourly rate of \$4, which is 13 times the minimum wage of India, and 3 times that of the Phillipines,[7] the two top countries of origin of workers from developing countries [14].

Regarding the potential ambiguity in web page segmentation due to different levels of granularity, and to study this phenomenon, every web page has been annotated by five independent annotators. Altogether, with 5,231 assignments, we collected 42,450 segmentations (encompassing 627,080 segments) for the 8,490 pages, which took 976 annotator hours at a total cost of about \$8,500.

Annotator Interface. Figure 4 shows the instructions given to the annotators, which include an animation that exemplifies the creation and adjustment of segments. Below the instructions, the screenshot was displayed on which the annotators had to draw segments as translucent blue rectangles just like in the animation.

The design of the annotator interface was optimized in pilot experiments for simplicity and physical ease. We first used a direct selection of DOM nodes to specify segments, but this interface required complex multi-selections and also confused annotators who lack knowledge in HTML. We hence asked the annotators to draw free-hand rectangles instead, requiring a subsequent step to resolve inaccuracies from the drawing and to map the annotations to the DOM. To ease drawing, we employed a click-move-click interaction, which allows the index finger to remain relaxed almost entirely, enabling fast work for a prolonged time. Though annotators could nest rectangles, this happened in less than 3% of annotations.



6:32 Info Anno Approve Reject internally Reject
15:29 Info Anno Approve Reject internally Reject
54:50 Info Anno Approve Reject internally Reject
2 31:12 Info Anno Approve Reject internally Reject
28:56 Info Anno Approve Reject internally Reject
40:24 Info Anno Approve Reject internally Reject
16:14 Info Anno Approve Reject internally Reject

Figure 5: Reviewer interface, showing one assignment per row: annotator ID as image; reference (blue boxes) and annotation (red frames) for the test page (rotated and scaled); number of comments; time taken; buttons to show comments and annotator information, to show annotations for non-test pages, and to approve or reject the assignment.

Reviewer Interface. Figure 5 illustrates the reviewer interface that we built in order to monitor annotation progress and quality.[8] To quickly check up on annotators, we introduced one test page in each HIT, for which the reviewer interface shows both reference and annotator segmentation. As test pages, we started with segmentations created by ourselves, and then iteratively integrated more test pages where annotators largely agreed. If an annotator segmented the test page badly, we inspected the other annotations and meta data to judge whether they spent effort on solving the task. If so, we still excluded the annotation from the dataset, but paid the annotator for their fair work ("internal rejection"). In order to gather especially good annotators, all annotators were limited to ten tasks until we reviewed their tasks, and could only continue if most were approved. In total, we approved 5,231 assignments, internally rejected 6,152, and openly rejected 540.

Fitting to DOM Nodes. In order to map the inaccurately drawn segment rectangles to DOM nodes, we treat each DOM node as part of the segment if at least a fraction θ_c of its visible area overlaps with the rectangle. We optimize θ_c so that the visible area of all DOM nodes of a segment—a multi-polygon—best matches the original rectangle in terms of the area F_1-score (cf. Figure 6a). The recall of 0.79 shows that 1/5 of the rectangles' area is not part of segments, which is sensible as (1) annotators tended to draw rectangles that are a bit larger than necessary for speed, and (2) a multi-polygon naturally provides a tighter fit to DOM nodes than one rectangle. The precision, on the other hand, is very high (0.94), indicating only a few cases where the rectangles were drawn a bit too small. By adding DOM nodes that were nearly contained in the rectangles, however, the number of empty segments, which contain no DOM node and would thus be discarded, drops from from 7% to just 2%. Figure 6b shows the distribution behind these averages: most multi-polygons match the drawn rectangle indeed accurately.

[7] https://en.wikipedia.org/wiki/List_of_minimum_wages_by_country

[8] Mechanical Turk does not include sophisticated reviewing interfaces, so that we used our MTurk Manager (https://github.com/webis-de/mturk-manager), an open-source project that assists requesters in carrying out large-scale crowdsourcing tasks.

Annotation Quality Assessment. Table 2 shows the annotation quality in terms of the agreement measure we developed in Section 4.3. Annotators largely agree which text nodes belong together, as indicated by the very high F_{B^3} (0.78) for *chars*. Indeed, the rather large difference between *pixels* (F_{B^3} of 0.65) and *edges* (both 0.73) shows that a significant portion of disagreement is due to a different segmentation of blank space (i.e., background), which is irrelevant for most downstream applications. Moreover, as a comparison of the values using $\max(P_{B^3}, R_{B^3})$ highlights, nearly all disagreement is due to annotators working at different levels of granularity, and not because of vastly different segmentations. We thus conclude that our dataset presents a high-quality resource for web page segmentation, and could even be extended to provide a hierarchical ground truth segmentation in the future.

5.3 Segmentation Fusion

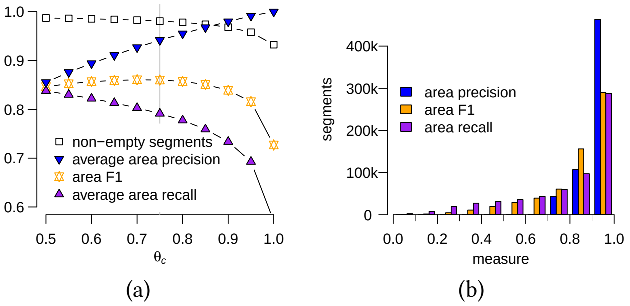
In order to fully utilize the wisdom of the crowd as well as to allow for easier evaluation of segmentation algorithms and training of learning-based ones, we fuse the five segmentations per web page into a single coherent ground truth as described in Section 4.3. We use *pixels* as the atomic elements to be in line with the annotation.

Figure 7 shows that fusing just elements that the majority of annotators put into segments (threshold of 3) reduces the number of *pixels* in the ground truth by 20%, but much less so the *edges* (6-7%), *nodes* (5%), and especially *chars* (2%). The annotators thus largely agreed on which elements are in a segment. The larger reduction for *pixels* is due to few annotators working at a more coarse level, for which segments naturally contain more blank space.

Figure 8a and b exemplify the fusion. Figure 8c compares the number of segments before and after the fusion for various θ_s. Roughly speaking, for $\theta_s = 0.9$ elements are put together in one segment if all annotators did so, whereas for $\theta_s = 0.1$ they are put together if any annotator did so. As is desirable, the figure shows that the distribution of the employed majority voting ($\theta_s = 0.5$) is also very similar to the original averaged distribution.

6 CONCLUSION

This paper revisits the task of web page segmentation, filling gaps that hindered the evaluation of generic web page segmentation algorithms. Unlike previous research, our evaluation framework does not focus on one of the various downstream tasks of web page segmentation. Instead it accounts for the different downstream tasks through a unified similarity measure for web page segmentations—well-founded in clustering theory—which can be used for tasks that focus on visual, structural, or textual elements. Moreover, we show how this measure can provide the basis for annotator agreement calculation, ground truth fusion, and segmentation quality assessment. This foundation is used to construct the Webis Web Segmentation Corpus 2020, a dataset that comprises 42,450 segmentations from human annotators for 8,490 pages from 4,824 sites. Our evaluation framework and this dataset allows for the first time to assess web page segmentation algorithms for different downstream tasks in a coherent fashion. Such a benchmark of common algorithms will be the logical next step and provide insight into task-specific strengths and weaknesses of the algorithms, thereby potentially revealing common issues of the algorithms and thus guiding future research.



(a) (b)

Figure 6: (a) Overlap of the annotated segments and DOM-based multi-polygons as well as fraction of non-empty segments at different thresholds θ_c, and (b) histograms of segments by the measures at the chosen θ_c of 0.75.

Table 2: Annotator agreement by type of atomic elements and pairwise measure within the agreement measure.

Agreement measure	Atomic page elements				
	pixels	*edges*$_{\text{fine}}$	*edges*$_{\text{coarse}}$	*nodes*	*chars*
F_{B^3}	0.65	0.73	0.73	0.74	0.78
$\max(P_{B^3}, R_{B^3})$	0.94	0.96	0.96	0.95	0.97

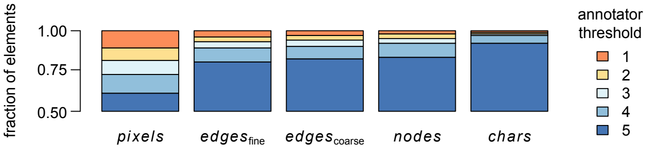
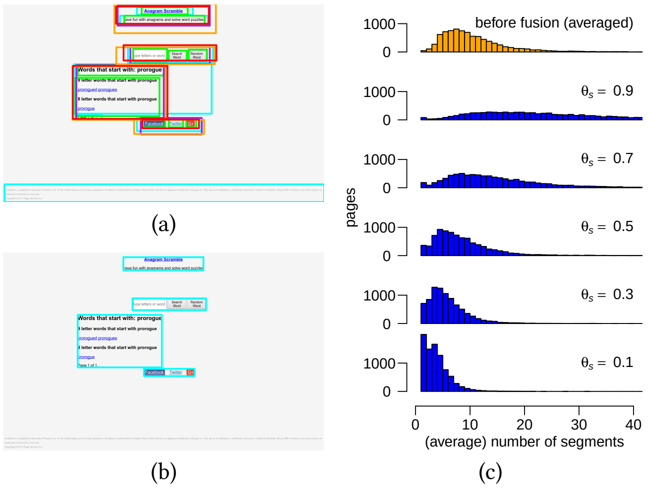


Figure 7: Fraction of atomic elements that are part of the ground truth for different annotator thresholds.



(a)

(b) (c)

Figure 8: Segmentations for the web page in Figure 3a, (a) by the five annotators (one color each), and (b) after fusion with $\theta_s = 0.5$. (c) Number of pages by the number of segments before and after fusion for various values of θ_s.

After web page segmentation, many downstream applications require a labeling of the segments. Therefore, an extension of our dataset with segment labels is a further step to continue this research. In the spirit of compatibility with as many downstream tasks as possible, a promising choice of segment labels is the function they fulfill on the web page [10]. Since such function labels carry a specific meaning, system developers can match such labels to the task at hand and then pick the corresponding segments, as well as evaluate with the extended dataset which algorithm performs best for segments that have the respective function.

REFERENCES

[1] M. Elgin Akpinar and Yeliz Yesilada. 2013. Vision Based Page Segmentation Algorithm: Extended and Perceived Success. In *Current Trends in Web Engineering - ICWE 2013 International Workshops ComposableWeb, QWE, MDWE, DMSSW, EMotions, CSE, SSN, and PhD Symposium, Aalborg, Denmark, July 8-12, 2013. Revised Selected Papers*. 238–252. https://doi.org/10.1007/978-3-319-04244-2_22

[2] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. *Information Retrieval* 12, 4 (2009), 461–486. https://doi.org/10.1007/s10791-008-9066-8

[3] Judith Jeyafreeda Andrew, Stephane Ferrari, Fabrice Maurel, Gael Dias, and Emmanuel Giguet. 2019. Web Page Segmentation for non-visual Skimming. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation, PACLIC 2019*. 423–431.

[4] Shumeet Baluja. 2006. Browsing on Small Screens: Recasting Web-Page Segmentation into an Efficient Machine Learning Framework. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*. 33–42. https://doi.org/10.1145/1135777.1135788

[5] Lidong Bing, Rui Guo, Wai Lam, Zheng-Yu Niu, and Haifeng Wang. 2014. Web Page Segmentation with Structured Prediction and its Application in Web Page Classification. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*. 767–776. https://doi.org/10.1145/2600428.2609630

[6] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Extracting Content Structure for Web Pages Based on Visual Representation. In *Web Technologies and Applications, 5th Asian-Pacific Web Conference, APWeb 2003, Xian, China, April 23-25, 2002, Proceedings*. 406–417. https://doi.org/10.1007/3-540-36901-5_42

[7] John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (June 1986), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

[8] Jiuxin Cao, Bo Mao, and Junzhou Luo. 2010. A Segmentation Method for Web Page Analysis Using Shrinking and Dividing. *IJPEDS* 25, 2 (2010), 93–104. https://doi.org/10.1080/17445760802429585

[9] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. 2008. A Graph-theoretic Approach to Webpage Segmentation. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*. 377–386. https://doi.org/10.1145/1367497.1367549

[10] Jinlin Chen, Baoyao Zhou, Jin Shi, Hongjiang Zhang, and Qiu Fengwu. 2001. Function-based Object Model Towards Website Adaptation. In *Proceedings of the 10th International Conference on World Wide Web* (Hong Kong) *(WWW '01)*. ACM, New York, NY, USA, 587–596. https://doi.org/10.1145/371920.372161

[11] Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. 2017. ICDAR2017 Competition on Recognition of Documents with Complex Layouts - RDCL2017. In *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017*. 1404–1410. https://doi.org/10.1109/ICDAR.2017.229

[12] Michael Cormier, Richard Mann, Karyn Moffatt, and Robin Cohen. 2017. Towards an Improved Vision-Based Web Page Segmentation Algorithm. In *14th Conference on Computer and Robot Vision, (CRV'17)*. 345–352. https://doi.org/10.1109/CRV.2017.38

[13] Michael Cormier, Karyn Moffatt, Robin Cohen, and Richard Mann. 2016. Purely Vision-based Segmentation of Web Pages for Assistive Technology. *Computer Vision and Image Understanding* 148 (2016), 46–66. https://doi.org/10.1016/j.cviu.2016.02.007

[14] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. 2018. Demographics and Dynamics of Mechanical Turk Workers. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. ACM, New York, NY, USA, 135–143. https://doi.org/10.1145/3159652.3159661

[15] D. W. Embley, Y. Jiang, and Y.-K. Ng. 1999. Record-boundary Discovery in Web Documents. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (Philadelphia, Pennsylvania, USA) *(SIGMOD '99)*. ACM, New York, NY, USA, 467–478. https://doi.org/10.1145/304182.304223

[16] David Fernandes, Edleno Silva de Moura, Altigran Soares da Silva, Berthier A. Ribeiro-Neto, and Edisson Braga Araújo. 2011. A Site Oriented Method for Segmenting Web Pages. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*. 215–224. https://doi.org/10.1145/2009916.2009949

[17] E. Bruce Goldstein. 2009. *Sensation and Perception* (8 ed.). Cengage Learning.

[18] Gen Hattori, Keiichiro Hoashi, Kazunori Matsumoto, and Fumiaki Sugaya. 2007. Robust Web Page Segmentation for Mobile Terminal Using Content-distances and Page Layout Information. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. 361–370. https://doi.org/10.1145/1242572.1242622

[19] Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[20] Johannes Kiesel, Fabienne Hubricht, Benno Stein, and Martin Potthast. 2019. A Dataset for Content Error Detection in Web Archives. In *18th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2019)*. ACM, 2.

[21] Johannes Kiesel, Florian Kneist, Milad Alshomary, Benno Stein, Matthias Hagen, and Martin Potthast. 2018. Reproducible Web Corpora: Interactive Archiving with Automatic Quality Assessment. *Journal of Data and Information Quality (JDIQ)* 10, 4 (Oct. 2018), 17:1–17:25. https://doi.org/10.1145/3239574

[22] 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 2008, Napa Valley, California, USA, October 26-30, 2008*. 1173–1182. https://doi.org/10.1145/1458082.1458237

[23] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, and Veljko M. Milutinovic. 2002. Recognition of Common Areas in a Web Page Using Visual Information: A Possible Application in a Page Classification. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*. 250–257. https://doi.org/10.1109/ICDM.2002.1183910

[24] Robert Kreuzer, Jurriaan Hage, and Ad Feelders. 2015. A Quantitative Comparison of Semantic Web Page Segmentation Approaches. In *Engineering the Web in the Big Data Era - 15th International Conference, ICWE 2015*. 374–391. https://doi.org/10.1007/978-3-319-19890-3_24

[25] Klaus Krippendorff. 1995. On the Reliability of Unitizing Continuous Data. *Sociological Methodology* 25 (1995), 47–76. http://www.jstor.org/stable/271061

[26] Xiaoli Li, Tong-Heng Phang, Minqing Hu, and Bing Liu. 2002. Using Micro Information Units for Internet Search. In *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 4-9, 2002*. 566–573. https://doi.org/10.1145/584792.584885

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision - ECCV 2014 - 13th European Conference*. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48 http://cocodataset.org/.

[28] Tomohiro Manabe and Keishi Tajima. 2015. Extracting Logical Hierarchical Structure of HTML Documents Based on Headings. *PVLDB* 8, 12 (2015), 1606–1617.

[29] D. Martin, C. Fowlkes, D. Tal, and J. Malik. 2001. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proceedings of the 8th International Conference on Computer Vision*, Vol. 2. 416–423.

[30] Chitra Pasupathi, Baskaran Ramachandran, and Sarukesi Karunakaran. 2012. Web Document Segmentation Using Frequent Term Sets for Summarization. *Journal of Computer Science* 8 (2012), 2053–2061. https://doi.org/10.3844/jcssp.2012.2053.2061

[31] Andrés Sanoja and Stéphane Gançarski. 2013. Block-o-Matic: A Web Page Segmentation Tool and its Evaluation. In *29ème journées "Base de données avancées", BDA'13*. 5.

[32] Andrés Sanoja and Stéphane Gançarski. 2015. Web Page Segmentation Evaluation. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, April 13-17, 2015*. 753–760. https://doi.org/10.1145/2695664.2695786

[33] Andrés Sanoja and Stéphane Gançarski. 2017. Migrating Web Archives from HTML4 to HTML5: A Block-Based Approach and Its Evaluation. In *Advances in Databases and Information Systems - 21st European Conference, ADBIS 2017, Nicosia, Cyprus, September 24-27, 2017, Proceedings*. 375–393. https://doi.org/10.1007/978-3-319-66917-5_25

[34] Robert R. Sokal and Charles D. Michener. 1958. A Statistical Method for Evaluating Systematic Relationships. *Univ. of Kansas Science Bulletin* 38 (1958), 1409–1438.

[35] Alex Spengler and Patrick Gallinari. 2010. Document Structure Meets Page Layout: Loopy Random Fields for Web News Content Extraction. In *Proceedings of the 2010 ACM Symposium on Document Engineering, Manchester, United Kingdom, September 21-24, 2010*. 151–160. https://doi.org/10.1145/1860559.1860590

[36] Srinivas Vadrevu, Fatih Gelgi, and Hasan Davulcu. 2005. Semantic Partitioning of Web Pages. In *Web Information Systems Engineering - WISE 2005, 6th International Conference on Web Information Systems Engineering, New York, NY, USA, November 20-22, 2005, Proceedings*. 107–118. https://doi.org/10.1007/11581062_9

[37] Jan Zeleny, Radek Burget, and Jaroslav Zendulka. 2017. Box clustering segmentation: A new method for vision-based web page preprocessing. *Inf. Process. Manage.* 53, 3 (2017), 735–750. https://doi.org/10.1016/j.ipm.2017.02.002