Multimodal Web Page Segmentation Using Self-organized Multi-objective Clustering

SRIVATSA RAMESH JAYASHREE, Indian Institute of Technology Patna, India
GAËL DIAS, Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, France
JUDITH JEYAFREEDA ANDREW, Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, France
SRIPARNA SAHA, Indian Institute of Technology Patna, India
FABRICE MAUREL, Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, France
STÉPHANE FERRARI, Normandie Univ, UNICAEN, CRISCO, France

Web page segmentation (WPS) aims to break a web page into different segments with coherent intra- and inter-semantics. By evidencing the morpho-dispositional semantics of a web page, WPS has traditionally been used to demarcate informative from non-informative content, but it has also evidenced its key role within the context of non-linear access to web information for visually impaired people. For that purpose, a great deal of *ad hoc* solutions have been proposed that rely on visual, logical and/or text cues. However, such methodologies highly depend on manually-tuned heuristics and are parameter-dependent. To overcome these drawbacks, principled frameworks have been proposed that provide the theoretical bases to achieve optimal solutions. However, existing methodologies only combine few discriminant features, and do not define strategies to automatically select the optimal number of segments. In this paper, we present a multi-objective clustering technique called MCS that relies on K-means, in which (1) visual, logical and text cues are all combined in a early fusion manner, and (2) an evolutionary process automatically discovers the optimal number of clusters (segments) as well as the correct positioning of seeds. As such, our proposal is parameter-free, combines many different modalities, does not depend on manually-tuned heuristics, and can be run on any web page without any constraint. An exhaustive evaluation over two different tasks, where (1) the number of segments must be discovered or (2) the number of clusters is fixed with respect to the task at hand, shows that MCS drastically improves over most competitive and up-to-date algorithms for a wide variety of external and internal validation indices. In particular, results clearly evidence the impact of the visual and logical modalities towards segmentation performance.

CCS Concepts: • **Information systems → Web searching and information discovery**; **Web interfaces**; • **Human-centered computing → Accessibility**; • **Computing methodologies → Natural language processing**; **Unsupervised learning**; **Reinforcement learning**.

Additional Key Words and Phrases: web page segmentation, multimodal early fusion, multi-objective optimization, self-organizing maps, evolutionary computation.

Authors' addresses: Srivatsa Ramesh Jayashree, Indian Institute of Technology Patna, 801103 Bihar, Patna, India; Gaël Dias, Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 6, Boulevard Maréchal Juin, Caen, France, 14000; Judith Jeyafreeda Andrew, Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 6, Boulevard Maréchal Juin, Caen, France, 14000; Sriparna Saha, Indian Institute of Technology Patna, 801103 Bihar, Patna, India; Fabrice Maurel, Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 6, Boulevard Maréchal Juin, Caen, France, 14000; Stéphane Ferrari, Normandie Univ, UNICAEN, CRISCO, Esplanade de la Paix, Caen, France, 14000.

Manuscript submitted to ACM

ACM Reference Format:
Srivatsa Ramesh Jayashree, Gaël Dias, Judith Jeyafreeda Andrew, Sriparna Saha, Fabrice Maurel, and Stéphane Ferrari. 2021. Multimodal Web Page Segmentation Using Self-organized Multi-objective Clustering. *ACM Transactions on Information Systems* 1, 1, Article 1 (January 2021), 49 pages. https://doi.org/10.1145/3480966

1 INTRODUCTION

Over the past years, web content creation has become more sophisticated, multimedia and modular [81]. In particular, content manager systems allow to place different modules on a web page in a way that appears coherent to the user. If carefully-placed, these modules cause most users to subconsciously segment the web page into coherent semantic regions, each one serving a specific goal (e.g., topic distinction, functionality highlighting, advertisement information).

Web page segmentation (WPS) aims at automatically identifying these coherent semantic regions, and can be defined as the process of breaking a large rendered web page into smaller regions, in which contents with coherent semantics are kept together [12]. Since the layout and the visual style of a web page are designed to facilitate content understanding, WPS should coincide with human's visual perceptive abilities and reflect the semantic coherence of a web page content.

There are several applications of web page segmentation. A classical ones consists in demarcating informative from non-informative content on a web page [15]. This (pre-)process has shown to improve the precision of web mining tasks like duplicate detection [15, 43], query expansion [12], and indexation [1]. There is a variety of other applications that benefit from web page partitioning. For instance, annotations for web images are more precise if they are extracted from the paragraphs the images belong to [11]. With proper segmentation, web pages designed for desktop screens can automatically be reconstructed for mobile devices [8, 20]. Web page accessibility can also be improved by the clarification of the document structure and clutter elimination [32, 47]. In information extraction, WPS may be used for the identification of data-intensive document sections [79] or individual data fields [59]. Finally, non-linear skimming strategies can be developed for visually impaired people by integrating WPS into a summarization pipeline [53].

Although WPS seems to be an easy task for humans, who unconsciously guess the outlines of a web page, automatically measuring the semantic coherence between contents in a modular document is a difficult quest. Indeed, the syntactic structure of a web page is designed for presentation purposes but without a clear description of the semantic relations holding between the different modules. As a consequence, the challenge of WPS is to model the semantic links that exist between the different information contents in a web page by taking into account layout, style and content properties, i.e. its morpho-dispositional semantics [55, 82, 83]. By modelling such multimodal semantics, partitioning strategies can then be proposed to segment web pages.

A wide range of supervised and unsupervised strategies have been proposed to tackle WPS. Supervised learning models have concentrated on delineating content from noise [77, 78, 80], eventually based on the findings of [18], who distinguish five block types: header, footer, left side bar, right side bar, and main content. Other attempts have focused on deciding whether some node elements in a graph-based web page representation should be considered as boundaries or not [9, 15]. Such models obviously require a large set of manually segmented web pages to take into account the wide spectrum of web design creativity. This certainly represents the main bottleneck of such methodologies. Indeed, it is unlikely that they can be used in real-world open-domain situations due to low performance rates for unseen web pages types.

To provide unrestricted web page segmentation, unsupervised strategies have been proposed [1, 4, 12, 15, 22, 37, 41, 43, 72, 84, 87]. Within this context, the most widely spread methodologies are based on *ad hoc* models, which highly rely on manually-tuned heuristics and may depend on parameters that need to be experimentally tuned

[4, 12, 37, 43, 68, 72, 84, 87]. To avoid such drawbacks, theoretically-founded methodologies have been proposed that either rely on graph-theoretic algorithms [15] or use classical clustering algorithms such as K-means, hierarchical agglomerative clustering and density-based clustering [1, 4]. Such principled solutions avoid the problem of defining a coherent set of heuristics, limit the intensive trial-and-error effort to combine these multiple heuristics, and are more likely to obtain global optima. Nevertheless, they usually cannot be executed at run-time, unless efforts are made to implement efficient solutions for the used clustering algorithms [7, 31, 33, 71, 76]. Another bunch of studies have been focusing on techniques borrowed from computer vision [22, 41]. Within this context, models to segment real-world photos [16] or digitized documents [56] are adapted to WPS. Results evidence low performance [41] confirming the experiments of [3]. Moreover, such strategies can be slow at run-time [41].

Modelling the multimodal semantics that links different web elements is the other pillar of WPS. For that purpose, different features have been studied. These can be classified into three main categories: visual, text and logical cues. Most methodologies rely on the DOM (Document Object Model) logical structure of the web page combined with the visual properties of its rendered version [12, 37, 72, 84]. While visual cues have shown to be the most important features [4, 12, 87], the DOM structure is helpful in a great deal of situations [37, 84], although it can be prone to errors due to uncontrolled page creation [87]. The first representative work to tackle text properties is [43], which evaluates text density as the only feature for WPS. As such, similarly to [37], it does not access to text semantics properties. This research direction is further proposed by [1] and followed by [9] for the supervised learning paradigm[1]. Surprisingly, although visual, logical and text properties have proven their discriminating power, only two studies have proposed to combine them all, i.e. [9] for the supervised case and [37] for the unsupervised case. While [9] proposes an integrated framework, where each cue is a feature for the learning process, [37] proposes an *ad hoc* two-step process, where clustering is first performed on visual and logical cues, and final clustering relies on text density similarly to [43].

In this paper, we propose to tackle WPS in a principled manner (as opposed to *ad hoc* strategies) by integrating visual, logical and text semantic properties (as opposed to text density) into a unsupervised (as opposed to supervised) model. As far as we know, this is the first attempt to formalize the problem of WPS with a well-known clustering algorithm, which includes all logical, visual and textual cues. Compared to the related work, our Multi-objective Clustering Segmentation (MCS) algorithm is parameter-free and does not depend on manually-tuned heuristics. As such, MCS is self-contained, does not depend on parameter fine-tuning, and can be run on any web page without any constraint. Moreover, it does not rely on pre-existing training data sets, which are difficult to build. The main contributions introduced in this paper are as follows:

(i.) We propose an evolutionary multi-objective learning framework called MCS based on the K-means algorithm that automatically finds the optimal number of clusters and correctly positions the initial seeds. For that purpose, an early fusion strategy is defined to combine all modalities into a single distance metric and four objective functions tackling individual modalities are concurrently optimized. A priority sorting strategy is then used to choose the best solutions on the Pareto front;

(ii.) A distance metric is formalized that includes visual, logical and text semantics indicators, thus following an early-fusion paradigm. For the visual cues, two indicators are taken into account: border-to-border distance and alignment distance between visual elements (aka. bounding boxes)[2]. For the logical features, two different distances are computed

[1]The paper mentions that text similarity is taken into account but this is not explained how. As such, the work is not reproducible and remains unclear for many of its features.
[2]Some experiments have been made based on background color distances but they were not conclusive.

based on the DOM structure: path length and logical dissimilarity (comparison of the common prefixes of the xpaths) between two bounding boxes. For text cues, text semantics is computed based on document embeddings [46];

(iii.) We propose the first attempt to deal with two distinct WPS tasks. On the one hand, we evaluate the performance of MCS over the classical WPS task, where the number of clusters is variable, i.e. the gold standard contains web pages segmented into different numbers of clusters. On the other hand, we test our algorithm on the specific task of WPS for non-visual skimming proposed by [3, 4]. Within this context, all web pages of the gold standard are segmented into exactly five clusters. Such a segmentation is motivated by the findings of [18] as well as the human perception capacities of concurrent speech [34, 38], combined with the Miller law [60]. Also, fixing the number of clusters can promote the automatic generation of formal invariants related to classical types of web page organization. As such, blind users may benefit from a more stable support, which may facilitate non-visual navigation in a transposed modality. It is worth noticing that the current work takes place within the TagThunder project[3] funded by the French Bank of Investment (BPI France)[4] that aims to provide skimming capacities to visually impaired people. As illustrated in Figure 1, WPS is one module of the overall architecture. As such, we show the versatility of MCS, which can easily adpat to different WPS situations unlike all other related works.

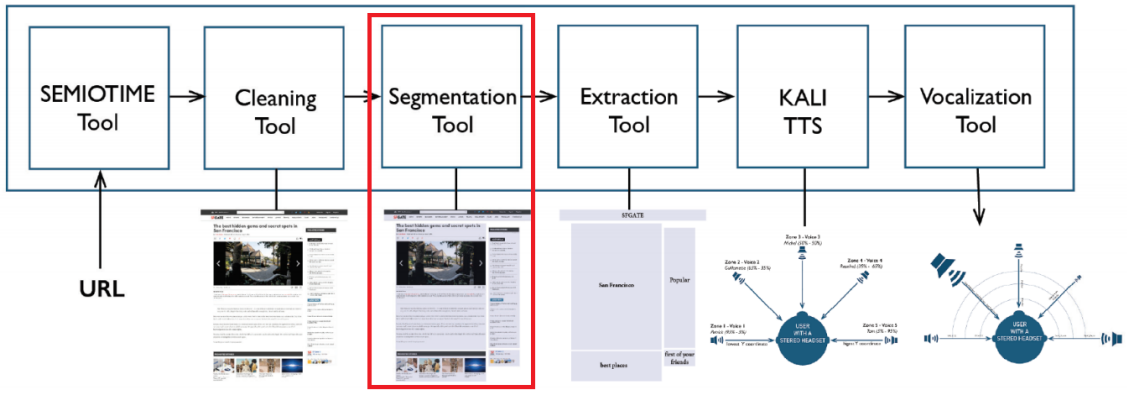

Fig. 1. Pipeline of the TagThunder project funded the French Bank of Investment. Image taken from [30].

(iiii.) We propose a strong evaluation setup compared to previous studies. In order to have a complete overview of the obtained results, we compute eight external validation indices (including BCubed metrics such as F_{b^3} [2]) and four internal validation indices. We also compare performance results and statistical significance with seven different baselines, including BCS [87], BOM [72], GE [4], and a set of alternative K-means baseline algorithms. We also propose two new implementations of the GE algorithm proposed by [4], that integrate a pre-clustering step based on the QT algorithm [36], which coherently upgrades the ideas of [4]. The underlying idea of such a strong evaluation setup is to propose the widest possible variety of evaluation metrics and comparable algorithms to better understand the impact of the MCS algorithm in the field of WPS. Over two gold standard data sets of 51 web pages (one for the unconstrained task and the other one for a segmentation into exactly 5 clusters), experimental results clearly show that MCS outperforms all related works for most of the external validation indices with statistical significance. Such results clearly evidence

[3]https://tagthunder.greyc.fr/
[4]https://tinyurl.com/9s2u2hvx

the benefits of combining visual, logical and textual modalities within a theoretically-founded framework, capable of dealing with different WPS situations.

In the next section, a complete overview of the related work on unsupervised web page segmentation is provided. In section 3, we present the problem formulation and describe the overall methodology. In section 4, we provide the reader with the learning setups that have been used to perform the experiments. In section 5, we present quantitative results for two distinct tasks, i.e., variable and fixed number of clusters in the gold standard data sets. Finally, we draw some conclusions and provide future research directions in section 6.

2 RELATED WORK

A wide range of research studies have been proposed to solve web page segmentation. Methodologies greatly vary with respect to (1) the learning strategy, (2) the processed data types, (3) the handled task, and (4) the evaluation procedures as described in Table 1. In order to better assess the evolution of this field of research, we propose to review the main related works[5] in this section. Note that we only focus our attention on unsupervised methodologies, which can directly be compared to our proposal.

	Algorithms	[12]	[84]	[15]	[43]	[1]	[72]	[87]	[4]	[37]	[22]	MCS
Cues	Visual	X	X	X	-	X	X	X	X	X	X	X
	Text	-	-		X	X	-	-	-	X	-	X
	Logical	X	X	X	-	X	X	-	-	X	-	X
Method	TD vs. BU	TD	TD	-	-	-	TD	BU	BU	-	TD	-
	AH vs. TH	AH	AH	TH	AH	TH	AH	AH	AH	AH	AH	TH
	ON vs. OF	ON	ON	OF	ON	OF	ON	ON	ON	ON	OF	OF
	PD vs. PF	PD	PD	PF	PD	PD	PD	PD	PD	PD	PD	PF
Evaluation	Manual	X	-	-	-	-	-	-	X	-	NA	-
	#EVI	-	1	2	2	1	5	2	-	3	NA	8
	#IVI	-	-	-	3	1	1	-	5	-	NA	4
	#RW	-	1	1	5	-	3	1	10	2	NA	7
	ET	1	-	1	1	-	-	-	-	-	NA	-

Table 1. Topology of unsupervised WPS strategies by types of cues, learning methods and evaluation frameworks. Note that TD (resp. BU) stands for Top-Down (resp. Bottom-Up), AH (resp. TH) for Ad Hoc (resp. Theoretical), ON (resp. OF) for On-line (resp. Off-line) clustering, PD (resp. PF) for Parameter-dependent (resp. Parameter-free) methodology, #EVI for the number of external valid indices used in the evaluation framework, #IVI for the number of internal validation indices, #RW for the number of tested related works, and ET for the number of tested external tasks. Note that NA stands for non available information.

2.1 *Ad hoc* Approaches

Ad hoc approaches stand for algorithms, which rely on specific heuristics and do not find their basis in theoretically-founded frameworks. Within this category, the VIsion-based Page Segmentation (VIPS) algorithm [12] is certainly one of the most accomplished solution. Its goal is to extract the hierarchical semantic structure of a web page, in which each node corresponds to a semantic coherent unit (aka. block). In particular, each node is assigned a value (called *degree of coherence* - DoC) to indicate the consistency of the content of the block based on its visual perception. The structure of the web page is obtained by combining the DOM structure and visual cues through three steps: block extraction, separator detection and content structure construction. This process is applied recursively. So, the web page is first segmented into several big blocks and the hierarchical structure of this level is recorded. For each big block, the same

[5]The reader can find a wide range of interesting references in [74] and [87].

segmentation process is carried out until a sufficient number of small blocks is obtained whose DoC values are greater than a pre-defined degree of coherence (PDoC). A manual evaluation over 600 web pages from popular sites listed in 14 main categories of Yahoo! directory showed that 93% of the web pages have their semantic content structures correctly detected. An extrinsic evaluation is also performed for query expansion that clearly shows that the vision-based web page content structure is very helpful to detect and filter out noisy and irrelevant information.

According to [84], since the visual architecture of a web page is to facilitate understanding, it should coincide with human's visual perceptive abilities and reflect the semantic coherence of its contents. Starting from the Gestalt theory, a psychological theory explaining human's perceptive processes, [84] developed a segmentation method (called GESTALT), that is based on four laws: closure, similarity, simplicity, and proximity. A layout tree is built from the DOM tree via a set of transformation hand-crafted rules to concisely describe the visual features of a web page. The closure step groups design patterns that are widely-accepted by humans (e.g., the list pattern). Following the similarity law, neighboring nodes under the same parent are grouped together if their similarity measure (based on the edit distance of the decoration node) value exceeds 0.7. Simplicity (here regularity) aims at finding repeated tree-like patterns that share some similarity. Through a greedy search algorithm, these patterns are recursively grouped together under a common parent. The proximity law stands for the separator detection process presented in [12], which aims to understand the visual structure of the web pages in terms of borders. The segmentation algorithm is tested over 60 web pages (3 from 20 different web sites) and evaluated based on Recall exclusively over a gold-standard data set[6]. GESTALT shows improved results over VIPS when a large number of clusters are to be discovered. But VIPS evidences best results for a small number of clusters as proximity plays the most important role in these conditions.

Unlike previous approaches, [43] presented the Block fusion (BF) algorithm, which can be defined as a purely text-based approach, that focuses on text density. The methodology can be executed at run-time as it is very fast because no complex document pre-processing is required. In particular, BF adopts the block growing strategy from image processing, where the decision when to combine (or fuse) two adjacent blocks is made by comparing them with respect to their text densities. Text density has the elegant property that no lexical or grammatical analysis is needed[7]. As such, a proper wrapping width (i.e., the slope delta threshold) is supposed to serve as a discriminator between sentential text (high density) and template text (low density). With respect to evaluation, the authors randomly picked 111 web pages coming from 102 different websites, and manually segmented them to create a gold-standard. Similarly to [15], they quantified the accuracy of the segmentation with two cluster correlation metrics: the Adjusted Rand Index (ARI), and the Normalized Mutual Information (NMI). Results show that BF performs significantly better than the graph-theoretic algorithm GCUTS [15], but no comparison is given with respect to VIPS or GESTALT.

Another strong baseline is proposed by [72], which extends the concepts of segmentation used for digitized document in the optical character recognition domain. In particular, they combine two popular approaches: the vision-based and the geometric layout models. The segmentation process of a web page is divided into three phases: page analysis, page understanding and page reconstruction. The page analysis phase consists in building a content structure from the DOM tree with the $d2c$ algorithm. The page understanding procedure uses this content structure and maps it to a logical structure via the $c2l$ algorithm, which depends on a granularity parameter pG. Finally, the page reconstruction phase gathers the DOM, the $d2c(DOM)$, and the $c2l(d2c(DOM))$ structures into a single structure that represents the segmented web page. To validate their study and compare it with different approaches, the authors built a manually-segmented test collection of 400 web pages crawled from dmoz.org Open Directory (25 web pages from 16 categories).

[6]Note that this is the first effort to build a gold-standard data set and provide automatic intrinsic evaluation.
[7]To the exception of tokenization.

In a paper about evaluation [73], they showed that their algorithm called Block-O-Matic (BOM) steadily improves over VIPS and BF for a set of specifically-tuned evaluation metrics, i.e. total correct segmentation, over-segmented blocks, under-segmented blocks, missed blocks and false alarms.

To avoid the dependency of DOM structures that might be error-prone, [87] propose to exclusively focus on the visual properties of a web page. For that purpose, they first use a rendering engine to get the smallest rendered elements (i.e. bounding boxes) of a web page. Segmentation is then performed using the Box Clustering Segmentation (BCS) algorithm that produces a flat set of segments of a given granularity. In particular, BCS is based on two different similarity metrics. The base similarity metric evaluates the visual similarity between two bounding boxes that are semi-aligned. Essentially, the base similarity is the arithmetic mean of three metrics (geometric distance, shape similarity and color similarity). The cluster similarity metric is used to express the similarity between two elements, where at least one of them is a cluster. A model based on clusters' inner similarity indicators is used for this effect, that builds on the idea of the degree of coherence in VIPS. Basically, the inner similarity is a mean value of base similarity that is calculated using the bounding boxes within a cluster. As a prerequisite of this representation, the model derives the direct neighborhood of each cluster from the direct neighborhoods of all the bounding boxes contained in that cluster. The idea behind BCS is to find the most similar couples of bounding boxes and to select them for merging based on a clustering threshold CT. With respect to evaluation, the authors created a specific data set that gathers 100 web pages of 8 different types from 5 news web sites. A semi-automatic approach was used to perform the annotation process. Results based on external validation indices (namely, ARI and F score) show mitigated conclusions as the accuracy of VIPS is better when processing structured pages, but BCS can provide better segmentations for less structured web pages.

Closely following the ideas of [87], [4] proposed the Guided Expansion (GE) algorithm for the specific task of non-visual skimming. Within this context, the segmentation process is constrained such that a fixed number of clusters should be discovered (here 5); the elements of a cluster should visually be connected; and all visual elements should be clustered. Within this context, two different strategies have been proposed. The first one aims at positioning the initial seeds of the bottom-up approach based on a reading strategy. The initial 5 seeds are arranged along the diagonal line of the web page, or along a F or Z line. These are called the D, F, and Z reading strategies. Once the seeds have been positioned, the GE algorithm sequentially assigns a bounding box to a given seed whether (1) it is the closest one in terms of border-to-border distance, or (2) it is aligned with it, or (3) it is the most visually similar in terms of CSS[8] properties. Note that once a bounding box is assigned to a seed, it becomes a cluster candidate. As such, when the visual cues are computed between a bounding box and its cluster candidate, the metrics are calculated between the bounding box and all the bounding boxes present in the cluster candidate. The second strategy combines an *ad hoc* density-like clustering algorithm [45] and the GE algorithm, and can be seen as two-step process. Here, the density-like algorithm first defines a coarse-grained clustering based on a given threshold, which defines some neighborhood of interest, and the GE assigns the remaining unclustered bounding boxes to their corresponding pre-discovered clusters. The evaluation results based on three internal validation indices over a set of 150 web pages from 3 domains show that GE is a good baseline, in particular if combined to the QT algorithm, when compared to a series of K-means algorithms. Note that they are the first authors to propose statistical tests to verify whether one algorithm is significantly different from some other one.

[37] are the first to propose a web page segmentation method that combines logical, visual and text semantic properties in a single model. For that purpose, they developed a two-stage methodology. First, a similarity model

[8]Cascading style sheets.

measures the similarity between bounding boxes by taking into account both visual and logical features. A geometric distance captures the distance between two bounding boxes, and the logical distance is the shortest path in the DOM tree. Both metrics are then combined linearly with some empirically-tuned parameter[9]. Based on the (visual and logical) similarity matrix of the web page contents, DBSCAN [28] is used to form big informative content blocks, and acts as a pre-clustering step, similarly to [4]. DBSCAN is a density-based clustering algorithm, that groups together elements that are closely packed together[10], and marks outliers that lie alone in low-density regions. The second step of the methodology consists in grouping together the pre-clusters that are positioned cohesively and show high textual density, following the ideas of [43]. To investigate the effectiveness of their methodology, the authors proposed an experiment on three data sets with respectively 70, 82 and 50 manually-segmented web pages over which they computed Precision, Recall and ARI external validation indices. Results showed that their methodology steadily improves over VIPS and BOM[11], with statistical significance.

2.2 Theoretically-Founded Approaches

Unlike *Ad hoc* approaches, some efforts rely on well-established clustering algorithms, the challenge being to adequately model the problem of WPS within theoretical frameworks. Within this context, [15] is the first approach to consider the problem of automatically segmenting web pages in a principled manner. The segmentation problem is cast as a minimization problem on a suitably defined weighted graph, whose nodes are the DOM tree nodes and the edge-weights express the cost of placing the end points in same/different segments. Based on logical and visual information, a single objective function is defined such that its minimization should result in a good[12] segmentation. Two concrete instantiations of this problem are proposed, one based on correlation clustering and another one based on energy-minimizing cuts in graphs. In particular, a supervised approach is defined to learn the edge-weights from manually labeled data. Through an intrinsic evaluation over a manually segmented gold-standard of web pages and an extrinsic empirical analysis for the specific task of duplicate web pages identification, results show that the energy-minimizing formulation (GCUTS) performs substantially better than the correlation clustering formulation. Comparatively to heuristic-based approaches, GCUTS can not be run online as the search space is combinatorial[13]. Moreover, no evaluation against heuristic-based solution is provided, that makes it hard to compare to state-of-the-art solutions. Finally, results show that supervised learning of edge-weights gives better results than unsupervised clustering, thus introducing an extra-step in the decision process, that might be repeated based on the collection at hand.

[1] test different clustering techniques, namely partitioning clustering, agglomerative hierarchical clustering and density-based clustering, over three distinct distance measures respectively based on DOM (logical), geometric (visual) and semantic (text) properties. To capture the logical distance between two bounding boxes, a metric based on two preconditions is defined: (1) adjacent sibling leaf nodes should have the same distances, and (2) the minimal distance of leaves belonging to different parents should be greater than the maximum distance of these leaves to their siblings. To account for the visual properties, a border-to-border distance is defined between two bounding boxes. To evaluate the textual semantic proximity[14], the knowledge-based similarity metric proposed by [21] is used. As such, instead of estimating a word-to-word lexical matching, the words of a bounding box are mapped to their corresponding concepts

[9] Note that this tuning is based on a training data set.
[10] Based on two parameters that need to be tuned.
[11] Note that they showed that BOM steadily outperforms VIPS over the three data sets
[12] As opposed to previous research, where different segmentations can be obtained depending on the granularity parameter.
[13] Some heuristics are proposed by the authors to limit the size of the search space.
[14] As opposed to [43], which uses text density.

in a knowledge base (here WordNet [61]) and a concept-to-concept accordance is computed between two atomic units. With respect to clustering strategies, (1) the K-medoid algorithm is tested for which the correct number of clusters is given *a priori*, (2) the single-linkage algorithm is used for the agglomerative hierarchical clustering, for which the given number of clusters is determined by cutting properly the dendogram, and (3) DBSCAN [28] is used as the reference for the density-based clustering, for which two parameters are considered empirically[15]. The framework is evaluated over a collection of 78 web documents from 8 different categories of the Yahoo! directory, and clustering performance is computed via the Dunn index for the internal and the rand statistics for the external validation indices. Unfortunately, each distance measure is applied alone and no solution is given to automatically retrieve the correct number of clusters. Moreover, results do not guarantee clear conclusions as the semantic property seems to play the most predominant role for the internal evaluation, but the logical view is the most discriminant for the external evaluation. Finally, the solution is language-dependent as measuring text semantic similarity requires the existence of external knowledge bases, which are not available for the vast majority of languages.

More recently, [4] proposed to test different versions of the K-means algorithm [51] for the specific task of web skimming for visually impaired people, where only 5 clusters must be discovered for any web page. Within this context, they exclusively rely on the border-to-border distance to perform clustering, and define a virtual bounding box for the assignment step of the algorithm. This novel idea allows to avoid the usage of the K-medoid as it is proposed by [1], but also enables to test different configurations of the K-means. In particular, they propose to adapt the K-means algorithm by exchanging the distance metric by a force metric simulating gravity attraction. The underlying idea was to test whether bigger blocks would attract (fuse with) smaller bounding boxes. The results based on the same evaluation setups discussed previously show that their *ad hoc* strategy GE outperforms all K-means versions. Moreover, the introduction of the force metric leads to descreased results, mainly due to the strategy used to select the initial seeds.

2.3 Computer Vision Approaches

A different research direction proposes to treat a web page as an image and use classical segmentation techniques borrowed from the computer vision field to achieve web page segmentation. Within this scope, one of the early initiatives is proposed by [22], which uses edge detection to find semantically significant edges. The algorithm relies on the image of the web page, and first calculates for each pixel the probability of a locally significant edge, which is based on how different the horizontal or vertical image gradients at the pixel level are from those of the surrounding pixels. Then, from these edge pixels, the algorithm composes horizontal and vertical line segments up to a maximum length of t_l. The algorithm then starts a top-down process with the entire page as one segment, and recursively splits the segments into two by choosing the vertical or horizontal line that is the most semantically significant, i.e. that has the most and clearest edge pixels. The algorithm stops if there are no semantically significant lines in a segment, or if a split would result in a segment with one side being less than s_{min} long.

[41] is certainly the most accomplished piece of work within this domain, by adapting two computer vision-based algorithms to the task of WPS. In particular, they tuned the hybrid task cascade model [16] from the MMDetection toolbox [17] by disabling the filtering step that only identifies segments containing real-world objects. No training step was required for this model as it is pre-trained on MSCOCO [49], a huge set of pre-segmented photos. The authors also adapted the convolutional neural network strategy proposed by [56], which is the state-of-the-art algorithm in segmenting digitized newspaper pages. In particular, instead of determining the position of text through optical

[15] Through the rand statistics.

character recognition, they used the positions of text nodes from the corresponding list of nodes that accompanies the gold-standard data set. Note that this model is supervised and needs to be trained. Results over the Webis-Webseg-20 data set [40] that comprises 8,490 manually-segmented web pages showed that in terms of BCubed metrics [2], VIPS is still a hard baseline to beat, and computer vision-based approaches can be efficient only if the handled task requires pixel-based segmentation (e.g. design mining). Note that the authors implemented an ensemble methodology combining 3 computer vision-based methods including [22], VIPS and HEPS [52][16]. But results were unsatisfactory.

2.4 Originality of the Proposal

The current baselines to perform WPS are all *ad-hoc* solutions that rely on hand-crafted rules and parameter tuning. Their success is mainly due to the fact that these algorithms can be run online. However, such methodologies suffer from different drawbacks. First, as they rely on parameter tuning, different segmentation granularities can be obtained depending on the definition of the given parameter. For instance, VIPS depends on the PDoC parameter that defines radically different hierarchical structures depending on its value. As such, exploratory studies must be performed to analyze, which value better suits the task at hand. Such a situation is clearly unsatisfactory as new research should be endeavored whenever the task changes. Second, these strategies highly depend on a huge number of *ad hoc* handcrafted rules, which do not guarantee to find "optimal" solutions. Moreover, such algorithms may not easily scale up as any modification within the rules may be incoherent with initial decisions. Third, a side effect of parameter tuning is the fact that many bounding boxes may end up unclustered. This situation was omnipresent for BOM and BCS during our experiments. Overall, [37] is certainly the most interesting related work as it builds on most previous findings of *ad hoc* strategies. Nevertheless, their solution relies on a set of 5 parameters that need to be tuned. The authors explain that future work clearly needs to be endeavored as current tuning is based on hard-coded heuristics that rely on human reading habits. Moreover, based on the tuned parameters, content elements are likely to remain unclustered. Interestingly, the authors also support the definition of more sophisticated text semantic features as paragraphs with similar subjects can be separated into different blocks based on just text density.

Less efforts have been focusing on proposing theoretically-founded frameworks, where WPS is defined through well-established clustering algorithms. This situation is mainly due to two different factors. First, principled strategies are usually slow and can not be run online. Second, the adaptation of clustering algorithms to the task of WPS is not straightforward. Indeed, problem formalization must carefully be defined to avoid non convergence issues. Nevertheless, theoretical solutions (1) avoid the well-known problem of defining a coherent set of heuristics that may be efficient for a wide range of web page types, (2) limit the manually intensive trial and error effort to combine these multiple heuristics, and (3) afford theoretical foundations that are more likely to obtain global minima as opposed to *ad hoc* methodologies that are inherently greedy, and may produce local minima. Within this context, previous works have not tackled the task completely. First, they mainly rely on the combination of a subset of modalities. Second, they usually do not provide theoretical solutions to automatically find the optimal number of segments, still relying on some threshold to define.

The other research direction that proposes to exclusively rely on the image-based segmentation of a web page to perform WPS, has shown mitigated results. From the different reported experiments, it becomes clear that VIPS outperforms the overall best computer vision options, unless the downstream task requires pixel-based segmentation, such as design mining [41]. In particular, [41] state in their conclusion that they "lay the foundation for the development

[16] A recursive rule-based approach that evidences very low results. As such, it is not presented in this paper.

of new approaches that may improve over the long-standing, yet heretofore unknown champion, VIPS". Thus, they clearly evidence the long way to go to reach the levels of BOM that steadily outperforms VIPS, within a wide range of studies. Moreover, best performing methodologies are supervised and need to be trained over gold standard data sets. Indeed, if unsupervised methodologies like [22] are adopted, results are weak and they still rely on parameters to be tuned.

In this paper, we propose the first model[17] that combines visual, logical and text semantic cues in a single theoretical framework. We call it Multi-objective Clustering Segmentation. In particular, we define a single distance metric that evaluates the visual, logical and semantic dissimilarity between two bounding boxes of a rendered web page. The clustering process relies on the K-means algorithm upon which a multi-objective optimization process automatically finds the "optimal" number of clusters and the "correct" positioning of the initial seeds based on concurrent maximization/minimization of four objectives. As a consequence, we propose a parameter-free framework that produces different optimal solutions of flat clusters, where each content element belongs to a unique cluster (i.e., there are no outliers). In order to investigate the effectiveness and the topology of MCS, we present the results of eight external and four internal validation indices for two different tasks: free web page segmentation (variable number of clusters) and constrained web page segmentation (fixed number of clusters) as in [4]. Indeed, as expressed in [63], most clustering evaluation metrics are biased towards a given specificity, and the correct understanding of the clustering process can only be achieved if a wide range of metrics are computed. Note that we present the first attempt to deal with two different tasks with the same theoretical framework. Finally, seven different configurations of related works are compared with MCS, namely BOM, BCS, GE (and its different versions), and K-means.

This piece of work is part of the TagThunder project [38] funded by the French Bank of Investment (BPI France)[18], which aims to provide first glance access to web pages in a non visual context (specifically visually impaired people). For that purpose, different modules are organized around the pipeline illustrated in Figure 1. From a given URL, the SEMIOTIME tool first builds a unique file that assigns to all bounding boxes its characteristics (e.g. xpath, coordinates, 122 CSS styles). Then, the cleaning tool processes the file by removing all non-visual bounding boxes (e.g. script, hidden elements, non displayed elements, null coordinates). The segmentation tool segments web pages into meaningful clusters after strategically defining the data points, i.e. the last HTML element of block type in the DOM tree (see Figure 7)[19]. Then, in each cluster, a set of relevant keywords are extracted that are further oralized by a text-to-speech engine. Finally, keyword oral signals are organized in a 2D or 3D space to reproduce the "cocktail party effect" [10]. It is important to notice that as we are following a theoretically-founded strategy to perform WPS, our solution can not be run online. Nevertheless, this constraint is not required by the TagThunder project, as the adopted business model relies on the offline pre-process of a given web domain. However, we are aware that this may be a hindrance to the development of new applications including WPS strategies. To overcome this situation, different research directions can be followed. On the one hand, quantum-inspired strategies such as QMEA [42], AQMEA [7] or other alike strategies [58] can be implemented. On the other hand, classical distributed architectures can also be implemented, which parallelize the different learning objectives [33]. New solutions over the cloud can also be tested [71]. Nevertheless, this remains out of the scope of this paper. Finally, the only constraint imposed by the TagThunder project is to work on real-world web pages written in French. A such, our developments have exclusively targeted the French language, although

[17] As far as we know.
[18] https://www.bpifrance.fr/
[19] For reading purposes, these last block elements are named bounding boxes or visual elements.

all presented methodologies can easily adapt to any language as no specific resource outside easy-to-get document embeddings [46] are needed.

3 MULTI-OBJECTIVE CLUSTERING SEGMENTATION (MCS)

Web page segmentation can be seen as a clustering problem, where bounding boxes should be structured coherently. Such clustering should satisfy two specifics as mentioned in [4]: (1) the elements of a cluster should be visually connected, and (2) all bounding boxes must be clustered (i.e., clustering is complete). For that purpose, different principled strategies [1, 4, 15] have been proposed. In [15], the segmentation process is presented as a minimization problem of a unique objective over a weighted graph. This framework allows the automatic finding of an "optimal" number of clusters, but it must rely on a supervised setting to simulate human-like segmentation as a huge number of clusters may be discovered in the unsupervised set up. [1] proposed different experiments based on K-means, agglomerative hierarchical clustering and density-based clustering (DBSCAN [28]). K-means [50, 51] is a well-known algorithm, which particularly suits the task of web page segmentation [4], but the number of K must be fixed *a priori*. Hierarchical clustering provides a tree structure instead of a flat set of clusters. As such, extra steps must be processed to define an "optimal" number of clusters, and such task is still an open question [39, 88]. Density-based clustering [28] has received some specific attention within WPS [37] as it can be tuned to simulate human-like processing based on its two parameters. However, such a process is task-dependent as different parameter values might be necessary for different web pages. Moreover, bounding boxes may remain unclustered [37], thus breaking the second principle mentioned above.

In order to overcome most drawbacks evidenced by previous theoretically-based approaches, we propose to build on the recent findings of [70] on K-means-based multi-objective clustering. As evidenced in [1, 4], K-means is a suitable algorithm for WPS as it provides a flat set of clusters, where no bounding box remains unclustered. The two well-know limitations of K-means are: (1) K must be fixed *a priori* (although it is usually not known in advance) and (2) the positioning of the seeds is random (which implies that results may depend on correct initialization). To deal with the first issue, traditional solutions [26, 64] require K-means to be executed multiple times with various values of K. The quality of the different partitionings is then measured with respect to some cluster validity index and the partitioning, which corresponds to the optimal value is selected. To deal with the second issue, alternatives to K-means such as K-means++ [6] and Global K-means [48] have been proposed that select specific positions of the initial seeds. Within K-means++, seeds are selected such as they maximize their inter-distance, while this is done incrementally adding one seed at a time for the Global K-means. Multi-objective methods [66, 70] propose an alternative to deal with both issues in a single framework following an evolutionary paradigm.

Existing traditional clustering techniques implicitly optimize an internal objective function, which may measure compactness, spatial separation, connectivity, density or symmetry between clusters. But in real-world situations, all these properties may not be captured using a single objective function. This is particularly true for WPS, where clusters should evidence different specifics depending on the viewpoint: visual, textual or logical. As such, the application of multi-objective optimization techniques that maximize/minimize different cluster validity indices has appeared to be a promising alternative [35, 69, 70]. In this paper, we propose a framework, which combines self-organizing maps (SOM) with a multi-objective differential evolution approach. This parameter-free K-means-based approach can automatically determine the number of clusters and consequently the optimal positioning of the seeds. For that purpose, a center-based encoding is used, where a set of cluster centers are encoded in the form of a chromosome. As such, both the number of clusters and the positionings of the seeds go through an evolutionary process that must maximize the overall quality of the subsequent partitioning based on concurrent objectives. Comparatively to existing strategies, the

number of K varies within a given range and the computation of the "optimal" number of clusters is based not only on a single objective function but on a set of objectives, which produce a set of "optimal" solutions on a Pareto front. Also, the evolutionary process allows to explore the search space of possible seeds positionings more exhaustively than K-means++ and Global K-means, thus maximizing the discovery of the globally optimal solution.

Comparatively to [70], where a document clustering solution in a unique continuous representation space is proposed, our Multi-objective Clustering Segmentation (MCS) algorithm must combine a set of discrete and continuous representations depending on the viewpoint (visual, textual or logical). As such, it can be seen as a multi-criteria clustering algorithm, where different objectives are simultaneously optimized. Moreover, different strategies are employed in terms of offspring reproduction. Firstly, we propose a specific pruning strategy based on SOM to ensure the diversity of the new population to be reproduced. As such, it is expected that the search space is more widely explored. Second, with respect to crossover, we define a methodology to reduce the neighborhood distance between chromosomes in the SOM from iteration to iteration so that convergence is boosted. Finally, no mutation is performed due to the problem representations constraints.

In the remainder of this section, we first set the problem formulation. Then, we briefly explain the different processing steps of MCS. Finally, we go into details about each step of the multi-view multi-objective clustering algorithm.

3.1 Problem Formulation

A web page is interpreted as an information set comprising of HTML elements, each one with its own set of attributes. These enriched HTML elements, rendered by a browser engine into bounding boxes[20] which form the leaves of the DOM structure, correspond to visual rectangle boxes. Each bbox has various attributes including but not limited to pixel coordinates, textual content, DOM path, and background color, which can be used as valuable cues to discover the inner layout structure of a web page. So, the task of web page segmentation can be formalized as follows.

- Given:
  - A web page with $\mathbb{N}_b$ number of bboxes $\mathbb{W} = b_1, b_2, ..., b_{\mathbb{N}_b}$ each one with its own textual, visual and logical features
  - A set of $\mathbb{N}_f$ objective functions $\mathbb{F} = F_0, F_1, ..., F_{\mathbb{N}_f}$ where each F_i evaluates how much the assignment of bboxes to a set of segments/clusters is optimized[21]
  - A range $[Kmin..Kmax]$ for the $\mathbb{N}_s$ number of segments/clusters to be discovered, i.e., $Kmin \leq \mathbb{N}_s \leq Kmax$
- Find:
  - An assignment $\mathbb{A} = A_0, A_1, ..., A_{\mathbb{N}_s}$ of the $\mathbb{N}_b$ bboxes such that
    - $\forall A_i \in \mathbb{A}, A_i = \{b_1^i, b_2^i, ..., b_{T_i}^i\}, |A_i| > 0$ (no cluster is empty)
    - $\bigcup_{i=1}^{\mathbb{N}_s} A_i = \mathbb{W}$ and $\bigcap_{i=1}^{\mathbb{N}_s} A_i = \varnothing$ (hard and complete clustering)
  - which simultaneously optimizes all objective functions in $\mathbb{F}$, i.e. $\mathbb{A}$ belongs to a Pareto optimal front.

3.2 Overall Evolutionary Framework

Given that there is no deterministic way to determine the "optimal" number of clusters K within the classical K-means algorithm, we adopt an evolutionary multi-objective optimization paradigm to explore the solution space (both in terms

[20] Referred to as bbox(es) in the remainder of this paper.
[21] Minimized or maximized, depending on the function.

of the number of clusters and the consequent positioning of the initial seeds). The overall process to cluster $\mathbb{N}_b$ bboxes into $\mathbb{N}_s$ segments based on the multi-objective version of K-means is detailed in Algorithm 1.

The Multi-objective Clustering Segmentation (MCS) algorithm starts by creating a random population of assignments, where each assignment consists of a set of random cluster centers whose size varies within a given range, i.e., $Kmin \leq \mathbb{N}_s \leq Kmax$. Within the evolutionary framework, a chromosome represents a given assignment $\mathbb{A}$ (and vice-versa), i.e. a set of cluster centers. A specific instanciation of the K-means algorithm[22] is then executed on each assignment/chromosome. Each chromosome is then evaluated by a set of $\mathbb{N}_f$ objective functions, where each one focuses on a specific viewpoint. A set of chromosomes is then selected based on the non-dominated sorting genetic algorithm (NSGA-II) [24] to take part in the offspring reproduction. Before reproduction proceeds, a self organizing map is trained to create a topographical map such that solutions which are similar in nature map to neurons next to each other, thus creating families of assignments. The SOM is used to prune the set of assignments in order to maintain an equilibrium in population size, while keeping a certain degree of diversity. The eventually-pruned selected set of assignments is chosen to run crossover, such that a new population is obtained. While the number of iterations is not reached, the new population is appended to the old population, and the evolutionary process repeats. Once the iterative process stops, a set of Pareto-optimal solutions is obtained and a single solution is chosen using priority sorting. The overall workflow is defined in Algorithm 1.

Algorithm 1: Multi-Objective Clustering Segmentation (MCS)

```
N ← random population of chromosomes {A^1, A^2, ..., A^|N|};
S ← ∅, selected population to reproduce;
Max ← maximum number of iterations;
Limit ← soft limit for population pruning;
while iteration number ≤ Max do
    Apply K-means on each A^i of N;
    Calculate objective functions in F over each A^i of N;
    Merge N with S;
    D ← At least top |N| selected solutions based on non-dominated sorting;
    Train SOM to group D into families;
    if |D| > Limit then
        S ← population in D pruned using neighborhood feature of SOM;
    else
        S ← D
    end
    N ← new population obtained from crossover operation on S using neighborhood feature of SOM;
end
Select best solution using priority sorting over the Pareto optimal front.
```

3.3 Chromosome Representation and Population Initialization

A chromosome encodes a set of different cluster centers, i.e., a possible assignment $\mathbb{A}$. As MCS attempts to determine the optimal set of cluster centers that can partition a web page appropriately, the number of cluster centers encoded in different chromosomes varies over the range, $Kmin \leq \mathbb{N}_s \leq Kmax$. For instance, to generate the ith solution of the overall population, a random number (K_i) is selected between $Kmin$ and $Kmax$, and these K_i number of initial cluster

[22] A specific distance metric is defined as well as specific update and assignment operators. These will be further explained in the paper.

centers are chosen randomly from the set of all bboxes contained in the web page. Note that lengths of input vectors (chromosomes) are kept equal, and therefore, variable length solutions are converted to some fixed length vectors by appending zeros at the end.

This set of chromosomes each one with a varying number of clusters forms the initial population N. In order to obtain a partitioning corresponding to a solution in the population, the update and assignment steps of K-means [50, 51] are executed on the whole data set considering the cluster centers encoded in the solution as initial cluster centers[23]. The discrete and continuous definition of the multi-criteria WPS problem explained in section 3 leads to the definition of a specific instanciation of K-means with proper update and assignment operators, which are defined in the following section.

3.4 Assign and Update Steps of *K*-Means

K-means clustering first assigns observations to cluster centers (assign step) and updates the cluster centers based upon the observations assigned to the respective clusters by using some form of an average (update step). Conventionally, the observations and the updated cluster centers belong to the same representation space. But in the current scenario, owing to the heterogeneous discrete and continuous attributes of bboxes, this can not be the case.

With respect to the assignment step, different metrics must be defined to take into account the different viewpoints (visual, logical and textual) in terms of distances between bboxes and cluster centers. This issue is detailed in section 3.4.2.

As for the update step, simple averaging may lead to illogical attributes (e.g., when coordinates are averaged, the sense of alignment is lost), and thus the concept of virtual bbox must be introduced. As the atomic units for WPS are bboxes, calculating the centroid of a cluster of bboxes should also be a bbox. However, such average bbox does not exist in the real data set. As such, we must conceptualize it. This is the virtual bbox. This issue is detailed in the following section 3.4.1.

3.4.1 Update Step. A virtual bbox v_k is defined by its pixel coordinates (top-left as (x_1, y_1) and bottom-right (x_2, y_2)), the set of bboxes that were assigned to it during clustering, and the continuous vector summarizing the textual contents of the bboxes assigned to it. Note that at initialization, each virtual bbox is one bbox of the web page. Then, at each iteration, the bboxes are assigned to virtual bboxes that are virtually conceptualized.

Let be a set of bboxes assigned to a virtual bbox v_k during the t^{th} iteration of K-means. Then, the coordinates of v_k for the next $(t+1)^{th}$ iteration are the average of the coordinates of the assigned bboxes (only top-left and bottom-right are taken into account). Since all bboxes are rectangular, the coordinates formed by taking the average of the assigned bboxes also form a rectangle. Thus, virtual bboxes are rectangular. The continuous text vector of a virtual bbox for the next iteration is formed by the concatenation of all the textual contents of the contained bboxes. This text is then transformed into a continuous space using Doc2vec [46].

In summary, at each iteration of K-means, the update step consists of building the virtual bbox by averaging the top-left and bottom-right coordinates over all bboxes allocated to it at the assign step, and then computing its continuous representation using the Doc2vec framework[24].

[23] Note that by taking a bbox as a cluster center, we are more considering the problem as a K-medoid strategy. However, in the following steps, virtual centers will be computed, so that we prefer to keep the mention of the K-means.
[24] https://radimrehurek.com/gensim/models/doc2vec.html

3.4.2 Assign Step. Once we have described the update step of the adapted K-means, we provide all the details of the assign step, which aim to allocate bboxes to a given virtual bbox at each iteration of the algorithm. In particular, we define three different families of distances (visual, logical and textual), as dissimilarities between virtual bboxes and bboxes. These can be evaluated using different viewpoints.

Visual Distances

Border-to-Border Distance: Similar to [4], we use border-to-border distance bbd as a measure of the geometrical distance between two rectangular bboxes. As the name suggests, border-to-border distance gives the closest distance between two bboxes, as illustrated in Figure 2. Since virtual bboxes (and bboxes) are rectangular in shape, the aforementioned distance is applicable to calculate border-to-border distance between a bbox b_i and a virtual bbox v_k, and it is noted as bbd_i^k. Let (x_1, y_1) and (x_2, y_2) be the respective top-left and bottom-right coordinates of the bbox b_i, and (x'_1, y'_1) and (x'_2, y'_2) be the respective top-left and bottom-right coordinates of the virtual bbox v_k, bbd_i^k be calculated as in Equation 1. Note that all distances are then normalized using min-max normalization.

$$bbd_i^k = \begin{cases} 0 & \begin{cases} ((x'_1 \le x_1 \le x'_2) \vee (x'_1 \le x_2 \le x'_2) \vee (x_1 \le x'_1 \le x'_2 \le x_2)) \wedge \\ ((y'_1 \le y_1 \le y'_2) \vee (y'_1 \le y_1 \le y'_2) \vee (y_1 \le y'_1 \le y'_2 \le y_2)) \end{cases} \\ \sqrt{(x'_1 - x_2)^2 + (y'_1 - y_2)^2} & \begin{cases} (x_2 \le x'_1) \wedge (y_2 \le y'_1) \end{cases} \\ y'_1 - y_2 & \begin{cases} ((x'_1 \le x_1 \le x'_2) \vee (x'_1 \le x_2 \le x'_2) \vee (x_1 \le x'_1 \le x'_2 \le x_2)) \wedge \\ (y_2 \le y'_1) \end{cases} \\ \sqrt{(x_1 - x'_2)^2 + (y'_1 - y_2)^2} & \begin{cases} (x'_2 \le x_1) \wedge (y_2 \le y'_1) \end{cases} \\ x_1 - x'_2 & \begin{cases} (x'_2 \le x_1) \wedge \\ ((y'_1 \le y_1 \le y'_2) \vee (y'_1 \le y_2 \le y'_2) \vee (y_1 \le y'_1 \le y'_2 \le y_2)) \end{cases} \\ \sqrt{(x_1 - x'_2)^2 + (y_1 - y'_2)^2} & \begin{cases} (x'_2 \le x_1) \wedge (y'_2 \le y_1) \end{cases} \\ y_1 - y'_2 & \begin{cases} ((x'_1 \le x_1 \le x'_2) \vee (x'_1 \le x_2 \le x'_2) \vee (x_1 \le x'_1 \le x'_2 \le x_2)) \wedge \\ (y'_2 \le y_1) \end{cases} \\ \sqrt{(x'_1 - x_2)^2 + (y_1 - y'_2)^2} & \begin{cases} (x_2 \le x'_1) \wedge (y'_2 \le y_1) \end{cases} \\ x'_1 - x_2 & \begin{cases} (x_2 \le x'_1) \wedge \\ ((y'_1 \le y_1 \le y'_2) \vee (y'_1 \le y_2 \le y'_2) \vee (y_1 \le y'_1 \le y'_2 \le y_2)) \end{cases} \end{cases} \tag{1}$$

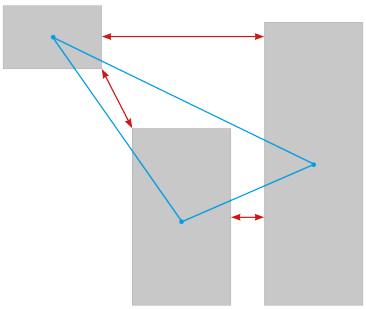

Fig. 2. Border-to-border distance (red) vs center-to-center distance (blue).

Alignment Distance: It has been shown that aligned parts of a web page share similar layout structures, which can be used as a valuable cue for WPS [4, 12, 84]. But alignment is a qualitative aspect that we need to quantify in order to fit to the K-means definition, the underlying idea being to define a distance metric in terms of alignment between two bboxes (either real or virtual).

Two bboxes are said to be aligned if their horizontal or vertical margin lines coincide. In particular, due to possible rendering errors or web page construction mistakes, we afford an error margin of 5 pixels[25]. For the iteration $t+1$, the alignment distance ald_i^k between a bbox b_i and a virtual bbox v_k is given in Equation 2, where bbd_i^j is the border-to-border distance between bboxes b_i and b_j, $IsAligned(b_i, b_j)$ equals to 1 if b_i and b_j are aligned and 0 otherwise, and C_k^t is the set of bboxes assigned to the virtual bbox v_k at iteration t.

$$ald_i^k = 1 - \max_{b_j \epsilon C_k^t, j \neq i} \left\{ \frac{IsAligned(b_i, b_j)}{bbd_i^j + 1} \right\} \tag{2}$$

It is important to remark that the alignment distance ald_i^k increases the tendency for aligned bboxes that are nearby to have higher alignment scores, and hence lower alignment distances for further bboxes, thus facilitating them to be in the same cluster. In other words, ald_i^k stands for the geometrically closest bbox aligned with the virtual bbox v_k as illustrated in Figure 3. Note that all distances are normalized using min-max normalization to fairly be included in a global distance.

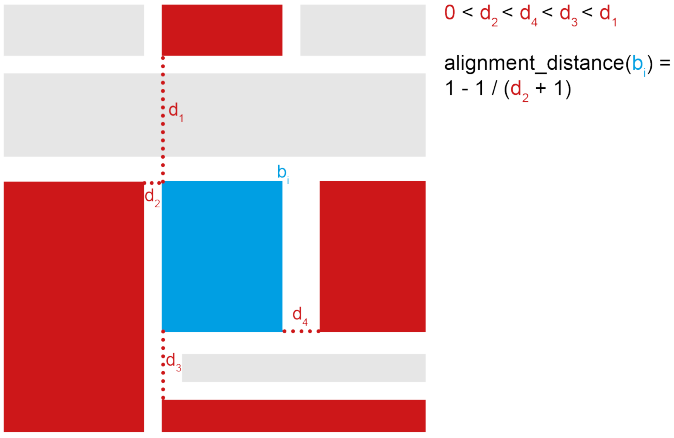

Fig. 3. Significance of alignment distance. Alignment distance is shorter between the blue bbox and the red one on its left, than with all the other red bboxes. Note that d_j stands for bbd_i^j, where i is the blue bbox and j are all possible other bboxes.

[25]This value has been set experimentally based on the used data sets. Note that the overall strategy does not depend on this parameter, which can eventually be set by default to 0.

Logical Distances

DOM Path Distance: Two similar bboxes can be disaligned or visually separated because of irrelevant contents[26], but may be close in the DOM tree structure. This situation may occur when there is an image element between two regions of text or when there is an advertisement in a section, for instance. So, following the ideas of [37], we use the xpath distance to measure the dissimilarity between the two bboxes in the logical DOM structure. Let l_i (resp. l_j) be the length of the xpath of bbox b_i (resp. b_j), i.e., the level or depth of the last element in the DOM tree structure of the bbox, and l_{ij} be the length of the common prefix in the xpaths of both bboxes, b_i and b_j, i.e., the level or depth of the lowest common ancestor in the DOM tree structure (an example is given in Figure 4). The path distance $pathDist_i^j$ between bboxes b_i and b_j is defined in Equation 3.

$$pathDist_i^j = l_i + l_j - 2l_{ij} + 1 \tag{3}$$

With respect to the DOM path distance between a bbox b_i and a virtual bbox v_k, we define the pd_i^k distance metric. At iteration $t+1$, pd_i^k is given by Equation 4, where C_k^t is the set of bboxes assigned to the virtual bbox v_k at iteration t. Note that all distances are normalized using min-max normalization, minimum and maximum being calculated for each web page. These are noted $\lceil pathDist_i^j \rceil$.

$$pd_i^k = \min_{b_j \epsilon C_k^t, j \neq i} \lceil pathDist_i^j \rceil \tag{4}$$

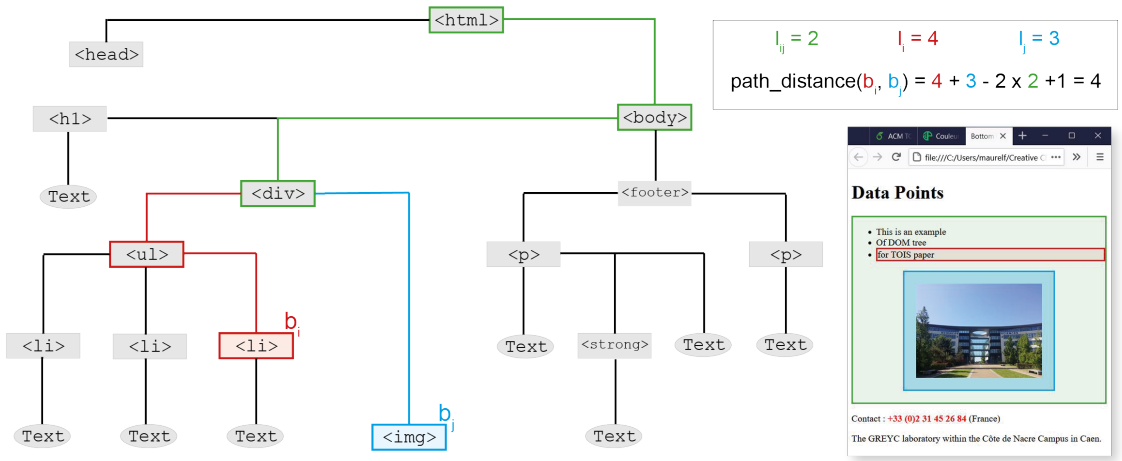

Fig. 4. An example of DOM path distance.

[26] Other reasons can be the cause.

DOM XPath Distance: Web pages may have nested complex DOM structures repeating multiple times. This is particularly the case for e-commerce web pages, where an image-text-button structure repeats itself multiple times. These bboxes though dissimilar in content are similar in intent, and we intend to leverage the similarity in structure using the xpath. Our purpose is illustrated with an example in Figure 5. The xpath similarity score $xpSim_i^j$ between two bboxes, b_i and b_j, is defined in Equation 5, where $\overrightarrow{b_i}$ (resp. $\overrightarrow{b_j}$) are the xpath vectors of b_i and b_j.

$$xpSim_i^j = \sum_{r=1}^{min(|\overrightarrow{b_i}|,|\overrightarrow{b_j}|)} \mathbb{1}_{\overrightarrow{b_i^r}=\overrightarrow{b_j^r}} \tag{5}$$

With respect to the xpath distance between a bbox b_i and a virtual bbox v_b at iteration $t+1$, we define the distance xpd_i^k as in Equation 6, where C_k^t is the set of bboxes assigned to the virtual bbox v_k at iteration t. Note that $\lceil xpSim_i^j \rceil$ is the normalized value of similarity using min-max normalization.

$$xpd_i^k = 1 - \max_{b_j \epsilon C_k^t, j \neq i} \lceil xpSim_i^j \rceil \tag{6}$$

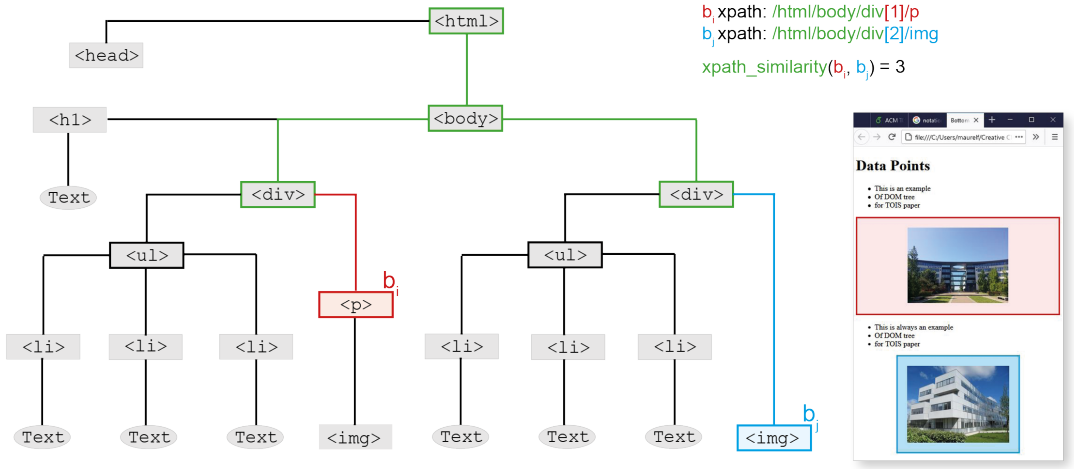

Fig. 5. An example of DOM xpath distance.

Textual Distance

A bbox may contain textual information and it can be interesting to measure the similarity between two different bboxes in terms of semantic content. Indeed, it is likely that words such as *cart*, *register* and *sign in* may refer to a single menu section. Recently, there have been a wide variety of methodologies to represent texts as continuous feature vectors, i.e., in some latent space [14, 46, 57]. Within this context, texts are represented as numerical vectors that can easily be compared. In the present work, we propose to use Doc2vec [46], but any related methodology could be used.

The textual content of a virtual bbox v_k can be defined as the sum of the textual contents of all its allocated bboxes in the previous iteration of the K-means algorithm. This content is then encoded by the Doc2vec framework as a numerical vector noted $\overrightarrow{v_k^{txt}}$. Similarly, a text continuous vector can be obtained for each bbox using the same Doc2vec procedure. Let this vector obtained for bbox b_i, be $\overrightarrow{b_i^{txt}}$. Thus, the normalized textual distance txd_i^k between a virtual bbox, v_k, and a given bbox, b_i, is calculated based on the cosine similarity measure, which is typically used for text similarity, and defined in Equation 7.

$$txd_i^k = \frac{1 - \frac{\overrightarrow{v_k^{txt}} \cdot \overrightarrow{b_i^{txt}}}{||\overrightarrow{v_k^{txt}}||.||\overrightarrow{b_i^{txt}}||}}{2} \tag{7}$$

Combining Multiple Distances and Defining the Assign Function

The distance between a virtual bbox v_k and a given bbox b_i can be evaluated from different viewpoints (visual, logical and textual), as illustrated in this section. However, in order to comply with the definition of the K-means algorithm, a unique distance metric must be defined[27]. This distance noted $dist_i^k$ is straightforwardly defined in Equation 8.

$$dist_i^k = \frac{1}{3}\left(\frac{(bbd_i^k + ald_i^k)}{2} + \frac{(pd_i^k + xpd_i^k)}{2} + txd_i^k\right) \tag{8}$$

Based on $dist_i^k$, the assign function of the adapted K-means can easily be defined as in Equation 9, where a bbox b_i is assigned to the cluster center b_m (at initialization), or the virtual bbox v_m if m is the closest center from all possible centers, b_k (at initialization) or v_k.

$$m = \min_k dist_i^k \tag{9}$$

The assign step of bboxes to virtual bboxes and the update step of virtual bboxes are repeated till convergence is obtained, and thus a grouping of bboxes into segments is obtained.

3.5 Objective Functions

To measure the correctness and magnitude of preference of a chromosome, various objective functions may be employed. The particularity of the multi-objective framework lies in the fact that it simultaneously optimizes different objective functions that may tackle different characteristics of a given assignment. This particularly suits the problem of WPS as partitionings can be evaluated in terms of visual, logical and textual features. Within this context, we define four objective functions. The Davies-Bouldin index [23] is used to define geometric and textual objectives, the Silhouette index [67] allows the definition of an alignment objective, and a heuristically-based objective function is defined to evaluate the proportion of logical cuts provided by a given assignment, i.e., the tendency of a given solution to cut logical HTML sequence structures such as lists <li> for instance, as suggested in [4].

3.5.1 Davies-Bouldin Index. The Davies-Bouldin index (DB) is a measure of compactness and separation of a given partition. DB is defined in Equation 10 for a partition of K clusters, which conditions constrain it to be symmetric and non-negative. DB is defined as a function of the ratio of the within cluster scatter, to the between cluster separation. As

[27] The study of multi-view K-means [13] can be an interesting research direction for future work.

a consequence, a lower value means that better clustering. In fact, DB is the average of the maximum ratio between the intra-cluster similarities S_i and S_j, and the inter-cluster distance M_{ij}.

$$DB = \frac{1}{K}\sum_{i=1}^{K} D_i \tag{10}$$

where

$$D_i = \max_{i \neq j} R_{1,j}$$

$$R_{i,j} = \frac{S_i + S_j}{M_{ij}}$$

S_i is a measure of scatter within the i^{th} cluster

M_{ij} is a measure of separation between the i^{th} and j^{th} clusters

3.5.2 DB-Border: Geometric Objective. We use the DB index to define the first geometric objective based on the border-to-border distance. Indeed, as previously mentioned in section 3, the elements of a cluster should be visually connected. Within this context, we define in Equation 11 the scatter function S_k^{bb} for the k^{th} cluster, whose bbox elements are $b_1, b_2, ..., b_{T_k}$ and its virtual bbox is represented as v_k.

$$S_k^{bb} = \left(\frac{1}{T_i}\sum_{i=1}^{T_k} (bbd_k^i)^2\right)^{1/2} \tag{11}$$

The separation function, M_{ij}^{bb}, between two clusters, C_i and C_j, is defined in Equation 12 as the border-to-border distance between their two virtual bboxes, v_i and v_j.

$$M_{ij}^{bb} = M_{ji}^{bb} = bbd_i^j \tag{12}$$

Note that this visual objective is required to be minimized during the evolutionary process to guarantee maximum visual connectivity between bboxes.

3.5.3 DB-Text: Textual Objective. Similar to the geometric objective, we use the DB index to define the textual objective in order to evaluate the overall semantic compactness and separation of a given partition. The underlying idea, is that clusters should demonstrate high inner coherence, and low outer consistency. Within this context, we make the assumption that the layout structure is closely related to the semantic content.

So, in Equation 13, we define the scatter similarity function S_k^{txt} for the k^{th} cluster, whose elements' text vectors are represented as $b_1^{txt}, b_2^{txt}, ..., b_{T_k}^{txt}$ and its virtual bbox's text vector as v_k^{txt}.

$$S_k^{txt} = \left(\frac{1}{T_k}\sum_{i=1}^{T_k} (txd_k^i)^2\right)^{1/2} \tag{13}$$

The separation function M_{ij}^{txt} between two clusters C_i and C_j is defined as the textual distance between their two virtual bboxes, whose semantic vectors are v_i^{txt} and v_j^{txt}. This situation is presented in Equation 14.

$$M_{ij}^{txt} = M_{ji}^{txt} = txd_i^j \tag{14}$$

Note that this semantic objective is required to be minimized during the evolutionary process to guarantee maximum semantic coherence of a given assignment.

3.5.4 SIA: Alignment Objective. It has long been shown that alignment plays a major role in WPS [4, 12, 72, 84]. As a consequence, an optimal partition should guarantee the maximum proportion of aligned bboxes within clusters, while evidencing a minimum percentage of aligned bboxes between clusters. To measure this phenomenon, we propose to build on the Silhouette index [67].

Alignment is a pairwise metric. As a consequence, we propose to use an objective function that facilitates the quantification of alignments in a pairwise fashion, in contrast to the *DB* index, which uses an average metric to summarize cluster separateness and compactness. The Silhouette index is a good candidate for that purpose, as it may compare, how good it is for a bbox to belong to a cluster as compared to the next best cluster.

For each bbox b_i assigned to cluster C_p with its virtual bbox v_p, and any other cluster C_k with its virtual bbox v_k, let

$$i_i = \sum_{j \in v_p, i \neq j} \mathbb{1}_{b_i \text{ is aligned } b_j}$$

$$e_i = \max_{k \neq p} \sum_{j \in v_k} \mathbb{1}_{b_i \text{ is aligned } b_j}$$

$$n_i = \sum_{j \in v_k, k \neq p} \mathbb{1}_{b_i \text{ is aligned } b_j} + i_i$$

$$sia_i = \frac{i_i - e_i}{\max\{n_i - i_i, n_i - e_i\}}$$

then, the Silhouette alignment index *SIA* is defined as in Equation 15, where $\mathbb{N}_b$ is the number of bboxes in a web page.

$$SIA = \frac{1}{\mathbb{N}_b} \sum_{i=1}^{\mathbb{N}_b} sia_i \tag{15}$$

Note that this alignment objective will have to be maximized during the evolutionary process to guarantee maximum inner alignment and minimum outer alignment of a given assignment.

3.5.5 CUTS: Number of Logical Cuts. In [4], authors proposed a new internal validation index. Based on manual evaluation, different experts evaluated negatively clustering results when logical constraints were broken, embodied by specific HTML tag sequences such as <li> items, <title> and the following paragraph <p>, <header>, <footer> or <nav> elements. As building logical objective functions based on the two previously defined distances, pd_i^k and xpd_i^k would result in erroneous indices due to their inner definitions, we propose to follow the same idea of [4] to take into account the logical viewpoint of a given assignment.

So, each time, one of these logical constraints is broken, this counts for one cut, and each web page is evaluated based on its overall number of cuts, the lesser, the better. As a consequence, this logical objective should obviously be minimized to guarantee adequate clustering. Note that we used the code provided by the authors of [4] to compute this objective function.

3.6 Non-Dominated Sorting and Pruning of Population

Once the adapted *K*-means has been run on the different chromosomes of the population, and each solution has been evaluated in terms of four different objective functions, the following step of the evolutionary process consists of

selecting the best individuals for the offspring reproduction. This step combines non-dominated sorting and population pruning based on self organizing maps.

Non-dominated sorting: Each chromosome has its own list of four objective values and non-dominated sorting is used to select the best solutions to reproduce over a set of Pareto-optimal fronts. For that purpose, we follow the same strategy as [70] and use the non-dominated sorting genetic algorithm (NSGA-II) [24]. In particular, NSGA-II sorts the solutions based on the concepts of domination and non-domination relationships in the objective functional space. For that purpose, it divides the solutions into a set of ordered fronts, each front containing a set of non-dominated solutions. $|D|$ top-ranked solutions are then selected from these fronts to take part in the offspring reproduction. In order to control the population size, if the number of selected chromosomes $|D|$ exceeds a pre-defined $Soft\text{-}Limit$, self organizing maps are employed to prune the population to a $Hard\text{-}Limit$.

Self organizing maps: Kohonen maps or self-organizing maps [44] are used to categorize the set of pre-selected solutions D into families in an unsupervised manner. SOM consists of nodes u (aka. map units) organized as a 2D grid, with each node occupying fixed cartesian coordinates, $z^u = (z_1^u, z_2^u)$.

Within the context of our work, each node in the SOM is structurally similar to a chromosome, each with a set of $Kmax$ virtual bboxes. Each virtual bbox of a given SOM map unit is made of the coordinates of the bbox and its content text vector. Note that contrarily to [70], each node is initialized with random positive real values.

Each chromosome of D must be assigned to a node of the SOM, and two chromosomes belong to the same family if they are assigned to the same map unit. As such, a chromosome i is assigned to the map unit u' that minimizes a given node-chromosome distance $d(i,u)$ as defined in Equation 16. Often, u' is called the winning node, or the best matching unit.

$$u' = \min_u d(i,u) \tag{16}$$

The distance $d(i,u)$ between a chromosome i and a node u is defined as follows. Comparatively to [70], where nodes and chromosomes share the same vector size, this may not be the case in our context. As a consequence, we must adapt the procedure to evaluate $d(i,u)$. Since a node has $Kmax$ centers (i.e., virtual bboxes) and a chromosome has any number of centers (i.e., virtual bboxes) ranging from $Kmin$ to $Kmax$, we adapt a greedy assignment policy to determine $d(i,u)$. Each virtual bbox v_i of chromosome i is paired with the closest possible node virtual bbox, v_u, in terms of dnc_i^u (a distance between two virtual bboxes defined in Equation 18) that hasn't been paired with any center of the current chromosome. This process is followed as long as there are no more centers left in the chromosome[28], and it is illustrated in Figure 6. So, the distance $d(i,u)$ between a chromosome i and a node u of the SOM is computed as defined in Equation 17.

$$d(i,u) = \sum_{|i|} \min_{v_i, v_u} dnc_i^u \tag{17}$$

For that purpose, we define dnc_i^u as the distance between a virtual bbox v_i of a chromosome i and a virtual bbox v_u of a node u, such that dnc_i^u is the summation of the border-to-border distance and the textual distance between both virtual bboxes[29]. This is formalized in Equation 18,

[28] This procedure does not necessarily yield to an optimal pairing, but the ability of SOM to learn leads to concluding results.
[29] Note that only these two distances can be randomly initialized, and as such, logical and alignment distances are not taken into account here.

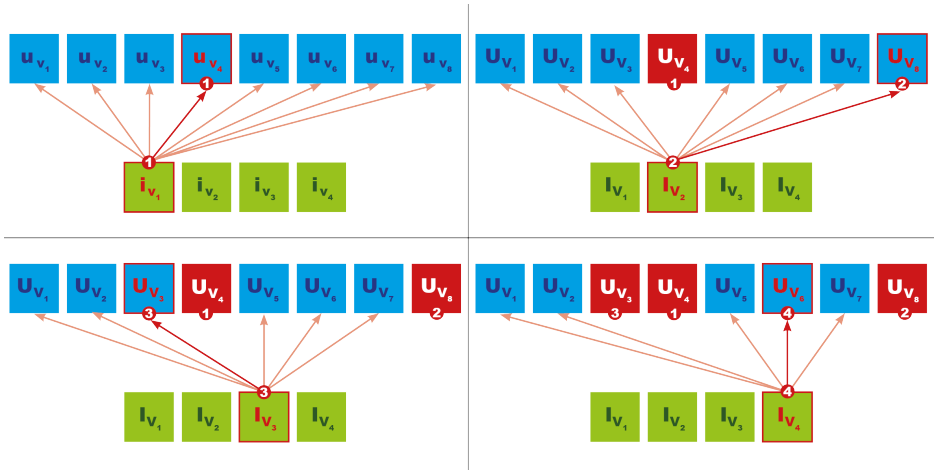

Fig. 6. Assignment of a chromosome of size 4 to a node of size $Kmax = 8$ of the SOM.

$$dnc_i^u = bbd_i^u + txd_i^u \tag{18}$$

The next step consists of computing the neighboring of the winning node u'. Since nodes are arranged on a 2D Grid with cartesian coordinates, a neighborhood U around u' can be defined as in Equation 19, where σ is a threshold that varies by iteration, and $\|.\|$ is the euclidean distance.

$$U = \{u \mid \|z^u - z^{u'}\| \leq \sigma\} \tag{19}$$

Note that σ is the neighborhood threshold defined in Equation 20 for each iteration t_{som} of a maximum T_{som} iterations of the learning process. The underlying idea is to gradually freeze the neighborhood to guarantee convergence.

$$\sigma = \sigma_{max}\left(1 - \frac{t_{som}}{T_{som}}\right) \tag{20}$$

Finally, the neighboring nodes u of the winning node u' must be updated to make them closer to each other. The corresponding centers (i.e., virtual bboxes) of the neighborhood nodes are updated as defined in Equation 21, where z^u (resp. $z^{u'}$) are the cartesian coordinates of node u (resp. u') in the SOM, x^u is the vector made of the average values for each feature across all the solutions assigned to the selected node u', and w^u is the feature vector of node u. Note that features for a given solution or node comprise of the coordinates of the virtual bboxes and the text vector. This is the batch version of the SOM algorithm.

$$\forall u \in U, w^u = w^u + \eta * e^{-\|z^u - z^{u'}\|} * (x^u - w^u) \tag{21}$$

Note that the learning rate η is defined in Equation 22 for each iteration t_{som} of a maximum number of T_{som} iterations of the learning process. The progressive limitation of the weights' updates also guarantees convergence.

$$\eta = \eta_{max}\left(1 - \frac{t_{som}}{T_{som}}\right) \tag{22}$$

All preceding steps are iterated until T_{som} is reached, so that convergence is achieved and the SOM gets stabilized.

Pruning of the population: If the number of selected chromosomes $|D|$ crosses a pre-defined $Soft\text{-}Limit$, the population must be pruned to a $Hard\text{-}Limit$. Comparatively to [70] who randomly select solutions in the population to perform pruning, we propose a methodology that guarantees the diversity of the chosen population to reproduce.

In particular, following a top-left bottom-right course strategy of the 2D grid, one chromosome is randomly selected (without replacement) for each node of the SOM representing a family of solutions until the $Hard\text{-}Limit$ is reached. This process maximizes the diversity of the solutions chosen for the offspring reproduction, thus optimizing the exploration of the search space.

3.7 Offspring Reproduction

Once the sufficient number of solutions have been selected for reproduction, crossover is performed to obtain a new population of potentially more performing chromosomes. For that purpose, we propose a randomized methodology that relies on the SOM. This differentiates from [70], who use a crossover operator of differential evolution.

A solution is randomly chosen in the population. Its counterpart for reproduction is randomly chosen from the set of solutions in its SOM neighborhood as defined in Equation 23, where t_{MOO} is the current number of iterations of the multi-objective optimization procedure that must be lower than the maximum number of iterations, T_{MOO}. Note that β_{max} is initialized with the same value as η_{max} in Equation 22. As such, as the optimization procedure proceeds, the neighborhood region decreases to ensure convergence.

$$\beta = \beta_{max}\left(1 - \frac{t_{MOO}}{T_{MOO}}\right) \tag{23}$$

All the individual genes of the two chromosomes are then stored into a unique bag, from which two new chromosomes are artificially built by evolution, as shown in Figure 7.

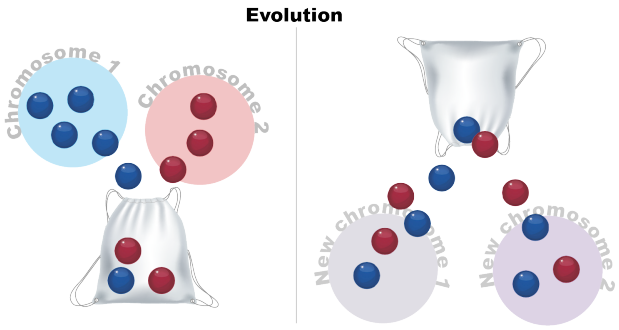

Fig. 7. Crossover bag strategy.

Two solutions are created by randomly assigning to two different bags a set of genes such that each bag contains between $Kmin$ and $Kmax$ genes and at least one gene from each initial chromosomes is contained in each bag. Thus, we guarantee the suitability of each solution and its evolution. Each bag is the new chromosome after reproduction.

The solutions that have reproduced are withdrawn from the set of possible solutions for reproduction, and the process continues until there are no more chromosomes to reproduce. Note that all solutions that cannot reproduce (i.e., the neighborhood is empty) are ignored.

3.8 Termination and Selection of Solutions

Once the new population has been created, it is merged with the pre-existing one in the next iteration of the optimization process, such that non-dominated sorting is executed again, and a new Pareto-optimal front is obtained. This process iterates until t_{MOO} equals to T_{MOO}, i.e., the termination condition is achieved.

When the optimization process ends, a Pareto-optimal front of non-dominated solutions is obtained from which a unique optimal solution must be selected. This issue is still an open problem as mentioned in [66]. In this paper, we propose a simple priority sorting strategy. The idea is to select the solution that maximizes a specific order of objective values. Let's note A, the Silhouette index alignment, T, the DB-Text, G, the DB-Border, and C, the number of Cuts. Then, if we decide upon the following combination of objectives, ACGT, the solution that maximizes A over all solutions will be chosen. In case of ties, the second objective to be minimized will be the number of cuts, and so on and so forth until the minimization of T. In this paper, for the sake of exhaustiveness, we will compare all 4! = 24 combinations, to understand the weight of each (visual, logical, textual) criterion for WPS.

4 EXPERIMENTAL SETUPS

In this section, we present the experimental setups that include the description of the data sets, the learning setups of the MCS, and the implementation details of the related works and the visual elements.

4.1 Data Sets

In order to test the MCS algorithm, we used the data set presented in [4], which consists of 53 web pages from 3 different domains (Tourism, E-Commerce, News), all written in French and part of the TagThunder project. Initially, 900 web pages have been automatically crawled from hub sites, i.e. 300 for each of the three domains. For each web page, a set of information was recorded that includes encoding, content manager system, image formats, to name but a few. Such process is illustrated in Figure 8. From this set of web pages, 30 items have been automatically selected for each domain so that the topological diversity is guaranteed for the gold standard data set. Indeed, web pages can greatly vary in terms of length, multimedia information, content manager system, and so on and so forth. And a strong evaluation set up should include all this diversity. After manual verification of the 90 web pages by three human experts[30], 23 web pages for Tourism, 12 web pages for E-Commerce and 18 web pages for News were selected for the gold standard data set. A subset of this gold standard data set is illustrated in Figure 9 to account for the diversity of the selected web pages.

In order to segment the gold standard data set, a specific annotation tool (WebSeg) has been implemented that allows to select a given number of clusters and assign visual elements to each defined cluster. Such a tool takes the form of a browser extension as can be seen in Figure 10.

There is no standard way to segment a web page, and it is easier to say that a web page is odd-segmented than to state the opposite. Indeed, WPS suffers from the same issues as most clustering problems, for which human judgment depends on different biases [29]. As a consequence, different clustering solutions may be judged "correct" although they are different. Nevertheless, within the creation of gold standards, some agreement must be attained over annotators.

[30] Three professors experts in the domain.

```
- <page>
  - <GlobalInfo>
      <id>52</id>
      <type>Tourisme</type>
      <addressWeb>http://www.toulouse-tourisme.com/</addressWeb>
      <isHomePage>Yes</isHomePage>
    </GlobalInfo>
  - <AccessbilityInfo>
      <nErrorAccessibility>6</nErrorAccessibility>
      <allTypesErrorsAccessibility>2 X Image button missing alternative text 4 X Empty link</allTypesErrorsAccessibility>
      <nImageButtonMissingAlternativeText>2</nImageButtonMissingAlternativeText>
      <nEmptyLink>4</nEmptyLink>
    </AccessbilityInfo>
  - <TechnologyInfo>
    - <FrameWorks>
        <frameWork>PHP</frameWork>
        <frameWork>DAV</frameWork>
        <frameWork>Shockwave_Flash_Embed</frameWork>
      </FrameWorks>
    - <Java-Script-Libraries>
        <Java-Script-Library>cufon</Java-Script-Library>
        <Java-Script-Library>Flash Object</Java-Script-Library>
        <Java-Script-Library>SWFObject</Java-Script-Library>
        <Java-Script-Library>MooTools</Java-Script-Library>
      </Java-Script-Libraries>
      <Widgets> </Widgets>
      <AggregationFunctionalities> </AggregationFunctionalities>
      <AdvertisingNetworks> </AdvertisingNetworks>
    - <Encodings>
        <Encoding>UTF-8</Encoding>
      </Encodings>
    - <MarkupLanguages>
        <MarkupLanguage>XHTML</MarkupLanguage>
      </MarkupLanguages>
    - <ImageFileFormats>
        <imageFileFormat>JPEG</imageFileFormat>
        <imageFileFormat>PNG</imageFileFormat>
      </ImageFileFormats>
      <Language> French </Language>
    - <ContentManagementSystems>
        <ContentManagementSystem>eZ Systems</ContentManagementSystem>
      </ContentManagementSystems>
      <EcommerceFrameworks> </EcommerceFrameworks>
      <ContentDeliveryNetworks> </ContentDeliveryNetworks>
      <Mappings> </Mappings>
      <AudioVideoMedias> </AudioVideoMedias>
    </TechnologyInfo>
</page>
```


Fig. 8. Example of the information gathered for a given web page.

Within the specific scope of our research, annotation has been manually performed by three human annotators[31] experts in the field.

For the first task that consists of segmenting web pages for non-visual access, all 53 web pages have been segmented with a fixed number of 5 clusters, as suggested in [4]. For that purpose, the three annotators have independently performed their own segmentation following the Gestalt theory as guidelines [84]. Then, all annotators jointly discussed their decisions for each of the web pages and decided on one consensus segmentation. Note that two web pages could not be segmented into 5 clusters as they were referring to error web pages.

Within the scope of this paper, we also want to deal with the more general case, where the number of clusters is not defined *a priori*. For that purpose, we performed a second round of annotations, where a given annotator had to define a segmentation with a variable number of zones, still following the Gestalt theory as guidelines. Note that no recommendation was given in terms of the number of clusters to be defined. Based on the experience acquired during the first annotation process, only one expert annotator segmented the set of 53 web pages with the objective to transcribe the major layout decisions made by the author when designing the web page. Then, all three annotators jointly discussed the proposed segmentations and agreed on some consensus. The results of the manual segmentation into a variable number of clusters are given in Table 2.

[31] Three professors.

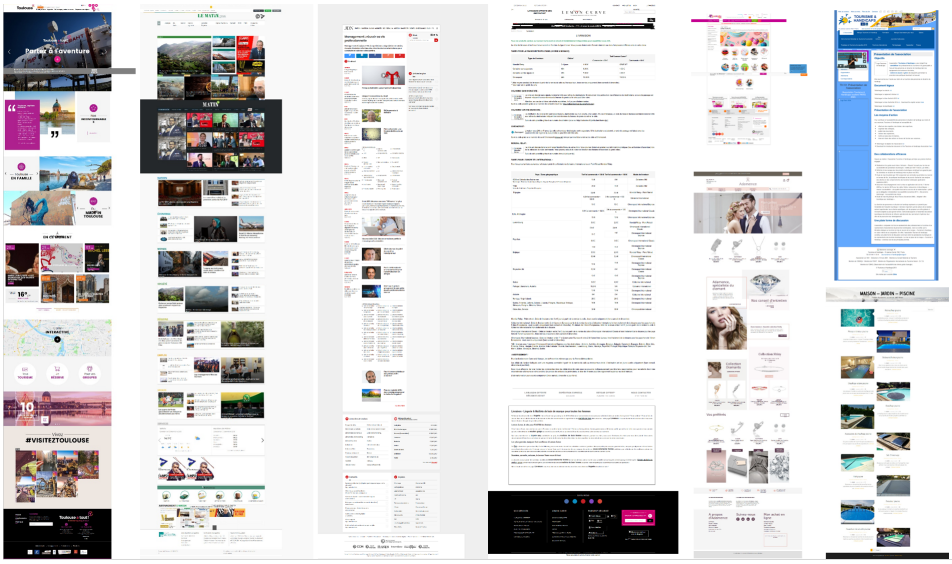

Fig. 9. Examples of the web pages included in the gold standard data set.

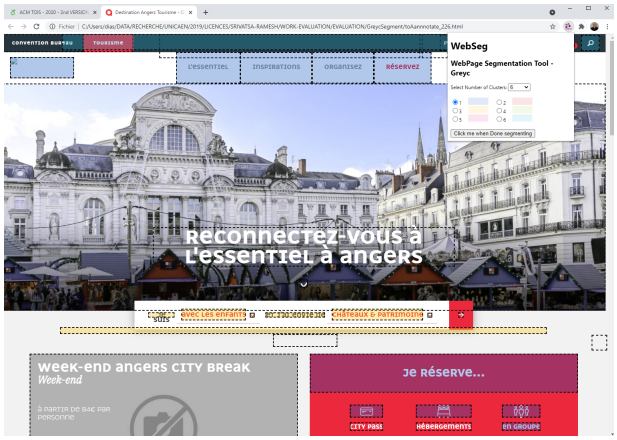

Fig. 10. WebSeg annotation tool. Extension for the Chrome browser.

Number of clusters	1	2	3	4	5	6	7	8
Number of web pages	2	0	4	7	16	14	4	6

Table 2. Distribution of web pages by number of clusters for the manual segmentation.

Interestingly, the majority class stands for the case where 5 clusters could be identified. This can be understood by the findings of [18], who had distinguished five main block types in web pages, namely, header, footer, left side bar, right side bar, and main content. This may also create a bias from the domains in observation. But, confirmed by the authors of [4], the choice of the initial 53 web pages was random, and as such, no bias was introduced in the selection of the

web pages. The maximum number of identified clusters is 8, suggesting that $Kmax = 8$, and the minimum number of clusters is $Kmin = 3$. Note that two web pages could not be segmented, i.e., only one cluster could be identified. These refer to error pages that were not taken into account for the final data set, which thus only contains 51 web pages.

4.2 Computation of Visual Elements

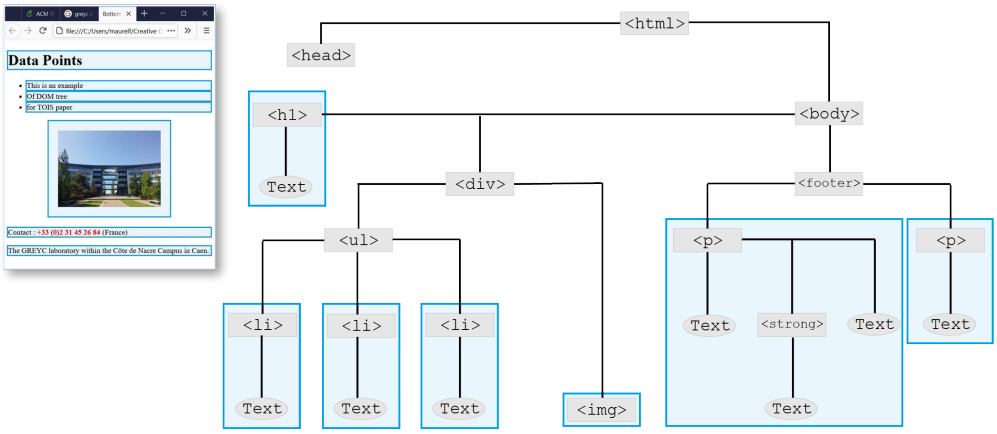

Fig. 11. Computation of the visual elements based on the DOM tree.

Before running the MCS algorithm, we need to define the visual elements (bboxes), which represent the data points[32]. For that purpose, different strategies can be developed [3, 87]. Here, we propose to define the data points as illustrated in Figure 11 and defined in section 4.3.3 of [3]. The top-down depth-first process of the DOM tree is defined in Algorithm 2.

It is important to note that some choices have been made to take into account the misuse of HTML elements for layout purposes, which are particularly common in real-world web pages. In fact, all strategies can be subject to discussion as there does not exist a standard way to compute visual elements. Within this set of experiments, we tried to define rules that allow the definition of small visual blocks. For example, the data values in table cells are considered as visual bboxes. As such, the clustering process may be hard as it must automatically reconstruct some elements that could easily be grouped using upper DOM structures.

Note also that a preprocessing step is necessary to prepare each web page so that visual bboxes can easily be computed. Such a preprocess consists first in computing the HTML rendering of the web page with the Selenium web driver[33] and the Mozilla FireFox browser[34]. The second step consists in adding additional information to each HTML element in the form of 3 attribute/value pairs by JavaScript[35] injection. These 3 attribute/value pairs are **data-bbox**

[32]Note that the notion of data point is classical in clustering, and in our case a data point refers to a given visual element to be clustered, i.e. a visible bounding box, also called bbox in this paper.
[33]http://www.seleniumhq.org/
[34]https://www.mozilla.org/fr/firefox/new/
[35]https://developer.mozilla.org/fr/docs/Web/JavaScript

Algorithm 2: Computation of the visual elements based on the DOM tree

```
/* List of non candidate Tags for Visual Elements (TVE).                    */
TVE ← [html, head, iframe, title, meta, link, script, style, strong, b, big, i, small, tt, abbr, acronym, cite, code, dfn, em, kbd,
  samp, var, a, bdo, br, map, object, q, span, sub, sup, button, input, label, select, option, textarea] ;
/* List of candidate Tags for Visual Elements (TVE).                        */
TVE ← [div, section, article, main, aside, header, footer] ;
visualElements ← ∅ ;
CNode ← first node of the DOM tree ;
repeat
    if CNode ∉ TVE then
        if CNode ∈ TVE ∨ CNode.style ∈ [display:block, display:inline-block] then
            if (child ∈ TVE ∨ child.style = display : inline) for all children of CNodes then
                visualElements.append(CNode);
            else
                if CNode contains exactly one child then
                    visualElements.append(CNode);
                end
            end
        else
            if CNode does not contain any child then
                visualElements.append(CNode);
            end
        end
    end
    CNode ← next node of the DOM tree;
until All tags of the DOM tree have been covered;
return visualElements;
```

for the coordinates and size of the HTML element, **data-xpath** for the path in the DOM tree and **data-style** for the 122 cascading style sheets (CSS) styles calculated by the browser. The format of the output files thus calculated is called HTML+. Finally, the last step aims at filtering those HTML+ elements which do not have any influence on the visualization of the web page for reasons of type, position, size or CSS style. Consequently, all HTML+ elements are given an additional attribute **data-cleaned** with the value **true** (if non visible) or **false** (if visible). This final output file gathers all the necessary information to perform the computation of the bboxes and constitutes the HTML++ format. An example of an HTML++ file is given in Figure 12 and this is possible to retrieve HTML++ files from any URL using the project's demonstration platform[36] by using the "cleaning" checkbox.

4.3 Learning Setups

Evolutionary Setup: The evolutionary process of the MCS algorithm contains a set of parameters that must be defined for learning. Contrarily to related works, these parameters do not interfere with the problem definition, as they do not change the shape of the building blocks nor the scanned region of the web page as evidenced in [37, 72, 87]. Instead, they allow a more or less adequate exploration of the searched space depending on their values. In this experiment, we focused on defining parameters' values that enable fast processing as WPS should ideally be run online.

[36] https://tagthunder.greyc.fr/demo/

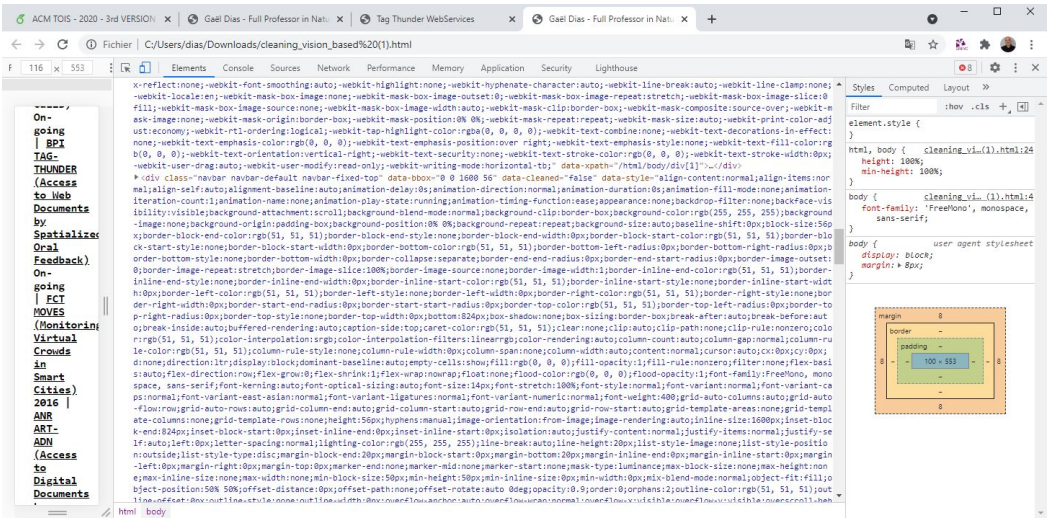

Fig. 12. HTML++ format of any web page to compute visual bboxes.

As a consequence, no particular tuning of the parameters was performed to obtain maximum performance over the data set, which anyway would consist in overfitting. All parameters' values are defined in Table 3.

Parameter	Value
Number of Segments	$[Kmin = 3 \,..\, Kmax = 8]$
Initial population count	15
MCS iterations	2
Soft-limit	30
Hard-limit	9
SOM number of nodes	$4 \times 4 = 16$
SOM iterations	5
SOM initial learning rate	0.5
SOM initial neighborhood distance	$2\sqrt{2}$
Doc2vec dimension	50

Table 3. Values of the parameters of the evolution process.

In Table 3, we mention that the number of iterations of the MCS algorithm should ideally be equal to 2, i.e. two iterations guarantee strong performance. The definition of this value is not random and it stems from an exhaustive analysis with respect to the number of iterations. In Figure 13, we present the average results in terms of F_{b^3} over the overall corpus, i.e. the 51 web pages of the gold standard data set for 10 iterations. The idea here is not to focus on performance results[37], but rather to understand the impact of the number of iterations. The linear regression clearly shows that improvements in terms of performance can be reached by iterating over the MCS, although at small pace.

However, each iteration of the MCS algorithm is computationally heavy as it includes the learning of the SOM, the application of all evolutionary operators, and the computation of the respective objectives. Following green AI good practices [75], the fine-tuning of frameworks should only be endeavoured if reasonable performance gain can be

[37] Such an effort is deeply detailed in section 5.

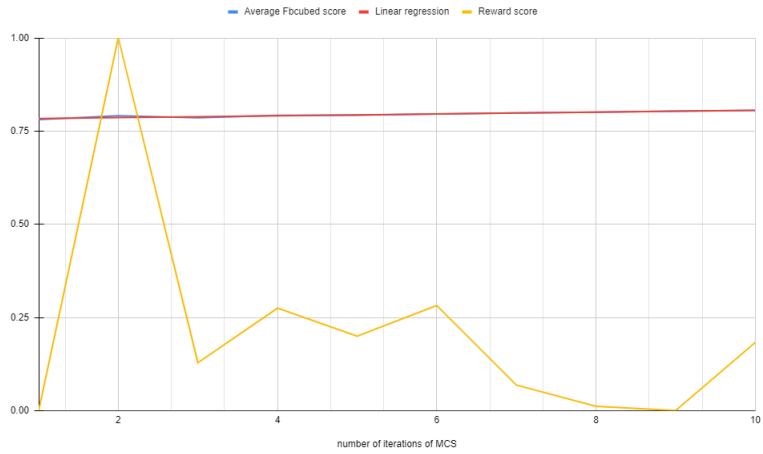

Fig. 13. Performance study by MCS iteration over the 51 segmented web pages with variable numbers of clusters. The reward function is based on F_{b^3}.

achieved (efficiency). Following the environmental trends, we define a reward function in Equation 24 that aims at estimating the efficiency of our framework instead of just its performance. Note that i stands for the iteration number and Performance is any gain function, where Performance^i stands for the performance value at iteration i.

$$\text{reward}(i) = \frac{\text{Performance}^i - \text{Performance}^0}{i} \tag{24}$$

From Figure 13, it is clear that the best efficiency is obtained for two iterations, where the F_{b^3} external validation index[38] is used as Performance function. As such, all results presented in this paper will be given for two iterations. Note, that better results can be obtained than the one presented in the next section for the MCS algorithm, but we prefer to follow the ideas of Schwartz et al [75] who state that "*progress will find more efficient ways [...] to reduce the computational expense with a minimal reduction in performance*".

Text Embeddings: To fully leverage the textual content of the bboxes, it becomes essential to quantify the similarity between them. As the textual contents of bboxes may range from a few descriptive words to long paragraphs, we need a mechanism that can handle the description of various text structures into some latent space. Within this context, different solutions could have been taken into account [14, 46, 86]. However, BERT [14] is a transformer-based framework, which fine-tuning is particularly difficult and power-dependent. Note that the pre-trained version of BERT for the French language [54] was not available at the time of the development of our solution. The recent multilingual version of the universal sentence encoder [86] was also not available at the beginning of the TagThunder project[39]. As a consequence, we used the open source implementation[40] of Doc2vec [46] to train a collection of 10,000 web pages randomly extracted from a set of 140 most visited (.fr) domains. This process results in a continuous n-dimensional representation space for web textual content, which is specifically required for the specific task at hand. As such, any text given to the Doc2vec framework can be represented by an embedding, i.e. a n-dimension semantic vector that

[38] It is explained in section 5.
[39] Implementations of more accurate text embeddings remains a future task.
[40] https://radimrehurek.com/gensim/models/doc2vec.html

is used to compute text similarity as defined in section 3.4.2. Note that for our experiments n was set to 50, which is a rather small dimension that may not capture all textual particularities, but goes towards the goal of maximum algorithmic efficiency.

4.4 Implementation of Related Works

In order to evaluate the performance of MCS, it is important to compare it to state-of-the-art implementations. In the domain of WPS, the development of practical solutions is a tedious task as it requires a large amount of engineering due to rendering issues. Moreover, many existing strategies rely on *ad hoc* solutions that heavily depend on heuristics, which are grounded on the correct processing of the HTML code. But, implementation details are usually omitted in the research papers, so that reproducibility is not guaranteed. As a consequence, we must rely on freely available frameworks or source codes.

For the sake of comparison, we used the plugin of BOM [72] that is made freely available by the authors[41]. As BOM depends on an input parameter that defines the size of the building blocks, we experimented different values on a small set of representative web pages, and finally decided that the best threshold value was 0.3. As a consequence, all 51 web pages have been segmented with the same value.

We also run BCS [87], as the authors kindly shared their executable code with us. Nevertheless, the provided code was not able to process all the 51 web pages due to rendering issues. Instead, only 13 web pages from our data set could be segmented. After some exchanges with the authors, we could not find an easy solution to fix the problem, and the only possible outcome was a new implementation of BCS, which is outside the scope of this paper. As a consequence, we will present the results of BCS based on this small set of samples only.

We also adapted the GE algorithm proposed by [4], as their code is available for research purposes. Indeed, although GE was implemented for a fixed number of clusters, i.e. $K = 5$, it can easily be tuned for a different number of clusters. Within this context, three different versions have newly been coded. The first version noted GE D. refers to the guided expansion algorithm [5], where initial seeds are placed on the virtual diagonal line of the web page. In the original code, five initial seeds were positioned on this line. We slightly changed the code so that any given number of seeds can be placed on the diagonal guaranteeing equal space between them. The second version is an extension of their algorithm called GE SP. [4], where a density-like clustering algorithm pre-processes the web page to find global large clusters before finalizing the clustering process with the guided expansion algorithm. Within this context, we propose to formalize this idea with the density-based algorithm QT[42] [36], as the initial methodology was an *ad hoc* proposal. In particular, different thresholds can be used with QT. To maximize performance, we experimentally tuned the threshold to be equal to 1/10 of the diagonal line dimension. We note this version GE QT. In the third version, the QT algorithm is not used as a pre-clustering step but rather as a methodology to find the initial positioning of the seeds. As such, QT is run over the web page, and the most central bbox of each cluster becomes an initial seed to run the guided expansion algorithm. This version is called GE QTC. With the new simple implementations of the GE[43], we want to provide the best possible configurations of GE to compete with MCS. However, it is important to note that any configuration of the GE algorithm needs to know *a priori* the number of clusters to be discovered. Indeed, there is no process to find the best value of K clusters.

[41] http://bom.ciens.ucv.ve/get-it/
[42] The Quality Threshold algoroithm.
[43] Note that we only consider GE QT. and GE QTC. as new implementations.

	C. Homogenity			C. Completeness			Rag Bag			C. size vs q.			Unbalanced		
Purity	0.71	0.79	✓	0.79	0.79	✗	0.56	0.56	✗	1.00	1.00	✗	0.96	0.96	✗
Inv. Purity	0.79	0.79	✗	0.79	0.79	✗	1.00	1.00	✗	0.69	0.92	✓	0.96	0.96	✗
F&M	0.47	0.49	✓	0.47	0.53	✓	0.61	0.61	✗	0.85	0.85	✗	0.95	0.94	✗
RandIndex	0.68	0.70	✓	0.68	0.70	✓	0.72	0.72	✗	0.95	0.95	✗	0.94	0.94	✗
Adj.RandIndex	0.25	0.28	✓	0.24	0.31	✓	0.40	0.40	✗	0.80	0.80	✗	0.79	0.79	✗
Jaccard	0.31	0.33	✓	0.31	0.36	✓	0.38	0.38	✗	0.71	0.71	✗	0.90	0.89	✗
F-measure	0.71	0.79	✓	0.79	0.79	✗	0.56	0.56	✗	1.00	1.00	✗	0.96	0.96	✗
P_{b^3}	0.60	0.69	✓	0.69	0.69	✗	0.49	0.56	✓	1.00	1.00	✗	0.93	0.95	✓
R_{b^3}	0.70	0.70	✗	0.71	0.76	✓	1.00	1.00	✗	0.69	0.88	✓	0.96	0.93	✗
F_{b^3}	0.64	0.69	✓	0.70	0.72	✓	0.55	0.71	✓	0.82	0.93	✓	0.94	0.93	✗

Fig. 14. Characteristics of external validation indices. Image taken from [63].

Finally, as it is clearly shown in [73, 74] that BOM outperforms VIPS and BF, we do not propose their implementations as comparative works.

5 EXPERIMENTAL EVALUATION

In this section, we present the results of MCS over two different tasks, i.e. the general framework, where the number of clusters is not fixed, and a task-oriented experiment, where a web page must be divided into exactly 5 coherent zones. As far as we know, this is the first algorithm that is tested in both situations. For that purpose, we use the set of 51 web pages segmented by our experts into a range of 3 to 8 clusters, and the same data set segmented into exactly 5 clusters.

In order to evaluate performance in the best possible way, we propose to calculate both external and internal validation indices. While intrinsic evaluation metrics measure how distant elements are from cluster to cluster (inter-cluster separation), and how close elements are within clusters (intra-cluster compactness), extrinsic metrics are based on comparisons between the output of the clustering algorithm and a gold standard usually built by human assessors.

With respect to internal validation indices, these correspond to the four objective functions implemented in MCS and discussed in section 3, i.e. the Davies-Bouldin index for border-to-border distance (DBV), the Davies-Bouldin index for text dissimilarity (DBT), the Silhouette index for alignment (SIA), and the number of cuts (Cuts). With regards to external validation indices, we propose to calculate a set of 8 different metrics that have all their own characteristics as stated in [2, 63], and illustrated in Figure 14. These external validation indices are Purity (P), Inverse Purity (INP), Rand Index (RI), Adjusted Rand Index (ARI), Jaccard Coefficient (J), Folks and Mallows (F&M), F-score (F), and B-cubed F-measure (F_{b^3}). Note that F_{b^3} is the only metric that tackles all the formal constraints stated in [2], and as such will be the main metric to support our conclusions. Note also that the formulas to compute the external indices are taken directly from [2].

When evaluating different clustering strategies, it is also important to verify if performance gaps are statistically relevant. For that purpose, we propose to use the non parametric Dunn Statistical test [27] that allows to test a set of solutions based on multiple comparisons using rank sums.

5.1 Variable Number of Clusters: General Framework

The first experiment consists in running MCS on the set of 51 web pages manually segmented into a variable number of clusters, i.e. K=[3..8]. As the evolutionary process of MCS provides a set of optimal solutions for each web page, a strategy must be defined to select one solution on the Pareto front. Within this paper, we propose to use the strategy of priority sorting explained in section 3.8, which consists in selecting the solution that maximizes a given objective at a time. So, if G stands for the border-to-border distance objective, T for the textual objective, A for the alignment objective and C for the number of cuts objective, a solution noted GTAC corresponds to the chromosome of the Pareto front that maximizes the G objective in the first place, then T in the second place, then A in the third place and finally C, in cases of ties. In order to be as exhaustive as possible, we will test all combinations of maximization, which consists of 24 (4!) configurations.

These configurations will be compared to BOM [72] and BCS [87] as explained in section 4.4. Moreover, the GE series of algorithms (i.e. GE D. [5], GE SP. [4], GE QT. and GE QTC.) will be tested in an "unfair" situation that consists of providing *a priori* the correct number of clusters to be discovered. Indeed, no process exists to find the optimal number of clusters. As such, these algorithms are tested in their best possible configuration, where the exact number of clusters is known in advance.

In order to evaluate the evolutionary process, we also provide the results of the classical K-means algorithm with $dist_i^k$, where the positioning of the seeds is random, and similarly to the GE series of algorithms, the correct number of K is given *a priori*. We also test a variant of the MCS algorithm, where the correct number of clusters is given and only the positioning of the seeds is optimized. These algorithms are noted K-GTAC, K-TGAC, K-AGTC and K-CATG, and correspond to the best configurations in terms of F_{b^3} for each initial objective.

5.1.1 External Validation Indices. Results of the 8 external validation indices over the set of 51 web pages manually segmented into a variable number of clusters are given in Table 4.

The best overall configurations of MCS are AGTC and AGCT, showing a maximum value of F_{b^3} equal to 77.2%. Note that AGTC and AGCT provide the exact same results as the same algorithms are selected for each web page. So, we will note AGXX the best performing strategy in terms of F_{b^3}. This confirms previous results [12, 84] that alignment and visual distance play a major role in WPS. The second best strategy, with almost the same performance (77.1% of F_{b^3}), is shared by the configurations that combine both alignment and number of cuts as primary objectives, i.e. ACGT and ACTG (noted ACXX). This result is particularly interesting as it confirms the recent findings of [4], that breaking logical structures negatively impacts WPS. But in any case, alignment seems to be the main discriminant feature for WPS. It is also worth noticing that the worst version of MCS in terms of F_{b^3} is the one that takes the solution that first optimizes the semantic textual content, i.e. TGAC, with F_{b^3}=65.5%, thus suggesting that it may be the weakest discriminant feature[44].

The second important result is that all configurations of MCS outperform all related works in terms of F_{b^3}, to the unique exception of GTAC and TGAC, when compared to the classical K-means. However, in this case, K-means receives the correct number of clusters to be discovered *a priori*, and as such is performed in an "unfair" situation. Compared to BOM, MCS in its best (resp. worst) configuration improves over 17.5 (resp. 5.8) points in terms of F_{b^3}. In parallel, it outperforms the best "unfair" implementation of GE series algorithms, i.e. GE QT., by 14.9 points, while the worst version of MCS shows improvements of 3.2 points. Comparative results with BCS cannot be taken as granted from the values in Table 4 as only 13 web pages could be segmented, but they give an idea of the performance of BCS, which was

[44] Of course, if its encoding is the correct one.

confirmed by a qualitative analysis. Indeed, it is clear that BCS provides worst results than BOM and consequently than MCS. This somehow confirms the results presented in [87], where BCS shows little or no improvement over VIPS.

When looking at the group control of MCS, i.e. K-GTAC, K-TGAC, K-AGTC and K-CATG, and the straightforward implementation K-means, it is rather astonishing to see that the evolutionary process performs better in terms of F_{b^3} when it automatically finds both the correct number of clusters, and the consequent initial positioning of the seeds. Indeed, the results obtained by AGTC are higher than the comparative K-AGTC, where the number of K is given as input and only solutions with a fixed number of clusters are present in the population. This suggests that the diversity of solutions with different values of K improves the correct positioning of the seeds. Moreover, when comparing K-AGTC to K-means, it is also clear that the evolutionary process provides better results when trying to find the optimal positions of the seeds, when compared to random initialization.

All comments given so far are based on the analysis of the F_{b^3} metric, as it is the one that satisfies most of the constraints present in clustering [2]. However, we also provide 7 other metrics that may satisfy subsets of these constraints. From the analysis of all metrics, it is clear that AGXX and ACXX configurations outperform all other ones, and very similar results are obtained between both these solutions. Within this context, Purity is an interesting case. Indeed, the control versions of MCS and K-means provide better results of Purity when compared to the MCS configurations. This can easily be explained as Purity penalizes non homogeneous clusters, and as such automatically over-evaluates results when large numbers of small clusters are provided. As the control versions of MCS and K-means know the exact number of clusters, they cannot be penalized if smaller numbers of clusters are provided by a given algorithm, which may be the case for all the configurations of MCS. This may also be true for Rand Index, Adjusted Rand Index and F-score. This issue will be discussed with the analysis of the internal validation indices in section 5.1.2.

	Algorithms	P	INP	RI	ARI	J	F&M	F	F_{b^3}
MCS	GTAC(*)	0.730	0.804	0.736	0.432	0.486	0.650	0.672	0.681
	TGAC(*)	0.726	0.798	0.718	0.407	0.459	0.632	0.654	0.655
	AGTC	**0.790**	**0.913**	0.823	0.619	0.630	**0.772**	**0.769**	**0.772**
	AGCT	**0.790**	**0.913**	0.823	0.619	0.630	**0.772**	**0.769**	**0.772**
	ATGC	0.776	0.907	0.807	0.590	0.613	0.758	0.755	0.757
	ATCG	0.776	0.907	0.807	0.590	0.613	0.758	0.755	0.757
	ACGT	0.789	**0.913**	**0.825**	**0.620**	**0.631**	**0.772**	**0.769**	0.771
	ACTG	0.789	**0.913**	**0.825**	**0.620**	**0.631**	**0.772**	**0.769**	0.771
	CGTA	0.749	0.862	0.770	0.508	0.550	0.704	0.719	0.725
	CGAT	0.749	0.862	0.770	0.508	0.550	0.704	0.719	0.725
	CTGA	0.762	0.880	0.782	0.532	0.563	0.719	0.731	0.736
	CTAG	0.762	0.880	0.782	0.532	0.563	0.719	0.731	0.736
	CAGT	0.781	0.898	0.807	0.583	0.604	0.751	0.756	0.760
	CATG	0.781	0.898	0.807	0.583	0.604	0.751	0.756	0.760
Related Works	BOM	0.609	0.852	0.620	0.258	0.410	0.595	0.600	0.597
	BCS(**)	0.651	0.760	0.593	0.206	0.375	0.558	0.571	0.569
	GE SP.	0.722	0.649	0.711	0.304	0.364	0.532	0.594	0.606
	GE QT.	0.670	0.798	0.650	0.282	0.395	0.584	0.603	0.623
	GE QTC.	0.633	0.825	0.600	0.238	0.395	0.583	0.579	0.601
	GE D.	0.682	0.647	0.706	0.295	0.373	0.536	0.597	0.576
	K-means	**0.815**	0.746	0.804	0.522	0.518	0.676	0.721	0.703
Contr. MCS	K-GTAC	0.811	0.733	0.777	0.465	0.472	0.642	0.701	0.693
	K-TGAC	0.771	0.751	0.757	0.435	0.470	0.636	0.683	0.675
	K-AGTC	**0.850**	0.818	**0.840**	**0.625**	0.618	0.759	**0.787**	0.768
	K-CATG	0.835	0.788	0.818	0.569	0.572	0.721	0.756	0.746

Table 4. Overall average segmentation results by external validation indices over the 51 web pages of the gold standard data set manually-segmented with a free number of clusters ($K = [3..8]$). (*) All combinations of selection give the same results. (**) Results have been computed using [87]'s toolbox, but some rendering errors were present and only 13 web pages could be segmented; thus results are shown only for these examples.

In Table 4, we provide average results for the 51 web pages. However, it can be interesting to understand how much values differ from web page to web page. For that purpose, we present the box plots for all the 8 external validation indices in Figure 15. Results confirm the conclusions drawn from Table 4, and also evidence that AXXX and CXXX configurations of MCS steadily outperform all other strategies, suggesting that alignment and the number of cuts are the most discriminant objectives.

5.1.2 Internal Validation Indices. Internal validation indices allow a more qualitative analysis of the clustering. Besides our four objectives, we also provide the average number of clusters (ANC) found by each configuration. Overall results are given in Table 5.

	Algorithms	DBV ↓	DBT ↓	SIA ↑	Cuts ↓	ANC
MCS	GTAC(*)	**0.76**	6.29	0.867	1.60	3.86
	TGAC(*)	2.75	**3.67**	0.873	2.46	4.52
	AGTC	1.83	5.41	**0.954**	0.50	3.56
	AGCT	1.83	5.41	**0.954**	0.50	3.56
	ATGC	2.12	5.16	**0.954**	0.70	3.58
	ATCG	2.12	5.16	**0.954**	0.70	3.58
	ACGT	1.86	5.36	**0.954**	0.46	3.52
	ACTG	1.86	5.36	**0.954**	0.46	3.52
	CGTA	1.28	6.37	0.912	**0.10**	3.40
	CGAT	1.28	6.37	0.912	**0.10**	3.40
	CTGA	1.85	5.12	0.921	**0.10**	3.48
	CTAG	1.85	5.12	0.921	**0.10**	3.48
	CAGT	1.57	6.14	0.942	**0.10**	3.40
	CATG	1.57	6.14	0.942	**0.10**	3.40
Related Works	BOM	68.0	**3.21**	0.925	1.92	3.35
	BCS(**)	4.45	3.43	0.796	2.41	3.58
	GE SP.	2.92	8.18	0.759	2.45	4.67
	GE QT.	5.05	3.36	0.864	3.10	4.69
	GE QTC.	15.5	4.45	0.869	3.41	4.69
	GE D.	7.67	3.97	0.694	5.67	4.69
	K-means	8.11	6.54	0.823	2.80	5.50
Contr. MCS	K-GTAC	0.95	6.75	0.816	2.26	5.50
	K-TGAC	46.0	3.94	0.865	2.74	5.50
	K-AGTC	6.23	5.91	0.932	1.28	5.50
	K-CATG	5.64	6.13	0.889	0.40	5.50

Table 5. Overall average segmentation results by internal validation indices over the 51 web pages of the gold standard data set manually-segmented with a free number of clusters ($K = [3..8]$). (*) All combinations of selection give the same results. (**) Results have been computed using [87]'s toolbox, but some rendering errors were present and only 13 web pages could be segmented; thus results are shown only for these examples.

It is interesting to note that drastically different clustering solutions can be found depending on the first objective to maximize in MCS. Logically, the first objective taken to select the MCS solution is the one with the best result over all configurations. For instance, GTAC is the algorithm that first optimizes the DBV index, and indeed, it shows the best value over all configurations with DBV= 0.76. With respect to ANC, TGAC is the configuration that most approximates the true average number of clusters that equals to 5.50, while all other configurations discover on average a smaller number of clusters. Nevertheless, this solution is the one with the worst results in terms of external validation indices, suggesting that clusters may be ill-formed.

Although AXXX and CXXX are the two best series of solutions when compared to related works in terms of external validation indices, they do not share similar clustering shapes, with non negligible differences over all internal validation indices. However, the best two configurations overall in terms of F_{b^3}, i.e. AGXX and ACXX, clearly evidence similar clustering behaviors. Interestingly, BOM provides a non-geometric clustering as its DBV index value is the highest

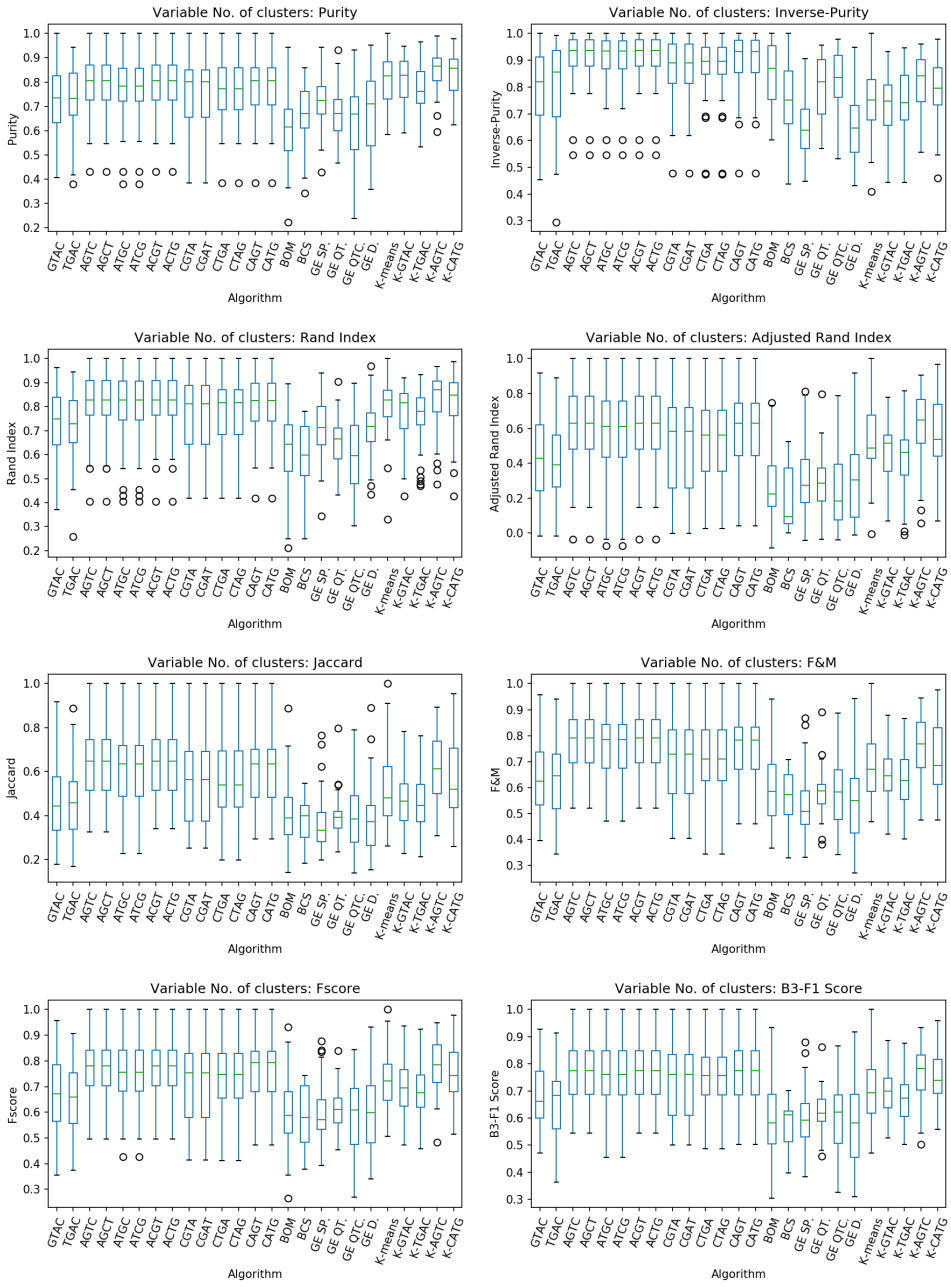

Fig. 15. Blox plot results for all tested configurations over the 51 web pages of the gold standard data set for all the external validation indices.

by a large margin over all tested algorithms[45], while it shows the best value for DBT, suggesting that it is clearly

[45]Note that DBV must be minimized to optimize clustering.

text-oriented, although it mainly depends on the DOM in its definition. With respect to BCS, results clearly show that the algorithm lacks in optimizing alignment with the second worst result overall in terms of SIA. It is also worth remarking that the GE series of algorithms can not guarantee that the exact number of clusters will be found. Indeed, due to their definition, it may be the case that a given seed is not assigned any bbox, thus justifying that ANC is not equal to 5.50. Moreover, the versions of GE that include QT pre-clustering, i.e. GE QT. and GE QTC., clearly outperform the original implementation of [4], i.e. GE SP., in terms of alignment and textual objectives, while down performing for the geometric and number of cuts objectives[46].

Nevertheless, most configurations of MCS, as well as BOM and BCS, provide a smaller number of clusters than the one that could be expected on average (i.e. 5.50), thus suggesting that future work is still needed to approximate the correct number of clusters, while maintaining high scores of external validation indices.

Finally, in order to understand how much the internal validation indices vary from web page to web page, we present the box plot results for the number of cuts in Figure 16. Indeed, as all web pages contain different numbers of clusters, showing the box plots for the other internal validation indices would not be interpretable. Results clearly show that the MCS configurations are more stable than the related works over the 51 web pages, with a small number of cuts for the best performing solutions in terms of F_{b^3}.

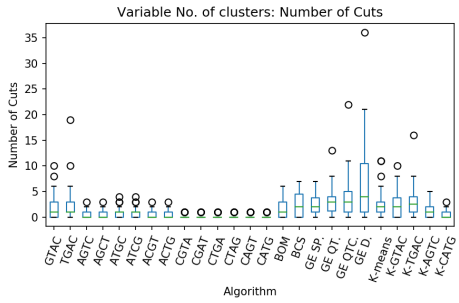

Fig. 16. Blox plot results for all tested configurations over the 51 web pages of the gold standard data set for internal validation index, number of cuts.

5.1.3 Statistical Analysis. In order to consolidate the results presented in Table 4, it is important to verify if performance gaps are statistically relevant. For that purpose, we show the Dunn Statistical test results in Table 6. Note that if two algorithms share at least one letter they are not statistically different.

Results clearly show the MCS configurations are statistically different from all related works, i.e. BOM, GE SP., GE QT., GE QTC. and GE D., for all tested evaluation metrics (both external and internal)[47]. However, there are no differences between AXXX and CXXX configurations for the external validation indices. Statistical difference is only observable between AXXX and CXXX for the number of cuts, which somehow confirms the superiority of AXXX over all other strategies. Note also that TXXX and GXXX configurations are statistically different from AXXX and CXXX for most tested situations. Interestingly, K-means is statistically different from the best versions of MCS, i.e. AGXX and ACXX, for all metrics except ARI, although it was provided with the correct number of clusters to discover.

[46] Further analysis is out-of-the-scope of this paper.
[47] BCS is not included as it does not contain the same number of tested web pages.

	Algorithms	ARI	J	F&M	F_{b^3}	Cuts
MCS	GTAC(*)	de	fg	fg	ef	fg
	TGAC(*)	cd	de g	def	d f	de
	AGTC	b	c	c	c	c
	AGCT	b	c	c	c	c
	ATGC	b	c	c	bc	c f
	ATCG	b	c	c	bc	c f
	ACGT	b	c	c	c	c
	ACTG	b	c	c	c	c
	CGTA	b d	c f	c g	bc e	b
	CGAT	b d	c f	c g	bc e	b
	CTGA	b e	bc	bc	bc	b
	CTAG	b e	bc	bc	bc	b
	CAGT	b	c	c	c	b
	CATG	b	c	c	c	b
Related Work	BOM	a	a g	a f	a d	e g
	GE SP.	a c	a	a	a d	a e
	GE QT.	a	a d	a d	a d	a
	GE QTC.	a	a d	a f	a d	a d
	GE D.	a c	a	a	a	a d
	K-means	b d	b ef	b e g	b f	a d

Table 6. Dunn Statistical test between all versions of MCS and all related works over the 51 web pages of the gold standard data set manually-segmented with a free number of clusters ($K = [3..8]$) for a subset of external and internal validation indices. (*) All combinations of selection give the same results.

5.2 Fixed Number of Clusters: Task-oriented Experiment

The second experiment consists in running MCS on the set of 51 web pages manually segmented into a fixed number of clusters, i.e. K=5, in the exact same settings as [4]. As such, the evolutionary process of MCS is limited to finding the optimal positioning of the seeds and the initial population of chromosomes only contains solutions with exactly 5 chromosomes. Note that in this case, the map units of the SOM are also vectors of size 5.

In particular, we test the best configurations of MCS in terms of F_{b^3} from the first experiment. So, for each objective, we take the best performing configuration thus leading to the following list of solutions to be evaluated: GTAC, TGAC, AGTC and CATG. Indeed, the goal of this experiment is to verify if MCS can adapt to a situation where the number of clusters is fixed, and an exhaustive evaluation of MCS in this situation is out-of-the-scope of this paper.

All versions of the GE series of algorithms are implemented to provide the most possible complete evaluation. As a consequence, we implemented GE D., GE Z., GE F. and GE SP. Note that GE Z. (resp. F.) stands for the version of the GE algorithm, where the initial seeds are positioned on a virtual Z (resp. F) line/shape over the web page. To complete the evaluation, we also implemented GE QT. and GE QTC. Finally, we implemented the classical K-means algorithm with $dist_i^k$, where the positioning of the seeds is random, being the control version of the MCS configurations in this context.

Overall results for external and internal validation indices are illustrated in Table 7. Note that only the number of cuts is given as internal validation index.

Performance values clearly evidence that MCS, with its AGTC configuration, outperforms all other clustering strategies for 7 external validation indices out of 8. In particular, it evidences an improvement of 5.4 (resp. 12.7) points in terms of F_{b^3}, when compared to the best (resp. worst) GE configuration, i.e. GE Z. (resp. GE QT.), and 6.4 points against K-means. Only the GE Z. and the GE QTC. algorithms illustrate better results for Inverse Purity, with a small margin, which tends to prefer more balanced partitions.

The second best configuration is embodied by CATG outperforming all other remaining solutions for 6 out of 8 external validation indices, but down performing to a reasonable margin of 2.8 points in terms of F_{b^3} with AGTC. These

Algorithms		P	INP	RI	ARI	J	F&M	F	F_{b^3}	Cuts
MCS	GTAC	0.813	0.733	0.783	0.487	0.493	0.657	0.702	0.697	2.08
	TGAC	0.751	0.742	0.742	0.410	0.449	0.616	0.668	0.659	2.88
	AGTC	**0.835**	**0.816**	**0.819**	**0.586**	**0.587**	**0.735**	**0.768**	**0.757**	1.12
	CATG	0.830	0.761	0.796	0.526	0.530	0.685	0.730	0.729	**0.41**
Related Works	GE SP.	0.762	0.652	0.737	0.366	0.398	0.567	0.619	0.633	1.94
	GE QT.	0.689	0.764	0.669	0.293	0.394	0.570	0.612	0.630	3.0
	GE QTC.	0.722	0.820	0.697	0.370	0.452	0.629	0.658	0.681	2.21
	GE D.	0.796	0.739	0.776	0.471	0.482	0.648	0.708	0.694	1.46
	GE Z.	0.747	**0.829**	0.753	0.470	0.520	0.686	0.711	0.703	2.04
	GE F.	0.715	0.795	0.707	0.373	0.450	0.624	0.662	0.667	1.88
	K-means	0.806	0.738	0.787	0.495	0.499	0.662	0.709	0.693	2.31

Table 7. Overall average segmentation results by external and internal validation indices over the 51 web pages of the gold standard data set manually-segmented with a fixed given number of clusters ($K = 5$). Only the best configurations with respect to F_{b^3} from the first experiment (i.e. $K = [3..8]$) have been taken into account for MCS.

results confirm our initial findings that the best configurations favour alignment and number of cuts, i.e. visual and logical cues, also in the case where the number of clusters is fixed.

However, contrarily to the first experiment, the GTAC and TGAC variants are not competitive against the best GE algorithm. In particular, GTAC gives similar results to the classical K-means, which can easily be understood by their close definitions. And, TGAC down performs compared to K-means, clearly evidencing that the semantic textual cue is the less discriminant feature overall.

It is interesting to note that the GE variants that do not include pre-clustering perform better than the ones with pre-clustering. This was the contrary for the case of a variable number of clusters. This may be explained by the existence of typical web pages, which follow some known content organisation as explained in [18], and for which a model-driven WPS can be useful.

With respect to the number of cuts, the only internal validation index studied in this experiment, the AGTC shows the second best result overall, only (naturally) overrun by the CATG variant, which clearly makes it the best solution overall.

In order to understand the complete sketch of the performance results presented in Table 7, we present the box plot for the 8 external validation indices in Figure 17, and the ones for the number of cuts in Figure 18. These illustrations confirm the conclusions drawn by the analysis of the average results presented in Table 7, with a clear positive impact of the AGTC and CATG variants for fixed-size WPS.

Algorithms		ARI	J	F&M	F_{b^3}	Cuts
MCS	GTAC	ef	cd	def	cd	de
	TGAC	de	bc	c e	bc	d
	AGTC	f	a	b	a	c
	CATG	c f	a c	b d	a d	b
Related Work	GE SP.	b d	b	a c	b	a c e
	GE QT.	b	b	c	b	d
	GE QTC.	b d	bc	de	cd	de
	GE D.	c e	cd	def	cd	c e
	GE Z.	a c e	a d	b f	cd	de
	GE F.	ab d	bc	a de	bc	de
	K-means	ef	cd	def	cd	a d

Table 8. Dunn Statistical test between best versions of MCS in terms of F_{b^3} and all related works over the 51 web pages of the gold standard data set manually-segmented with a fixed given number of clusters ($K = 5$) for a subset of external and internal validation indices.

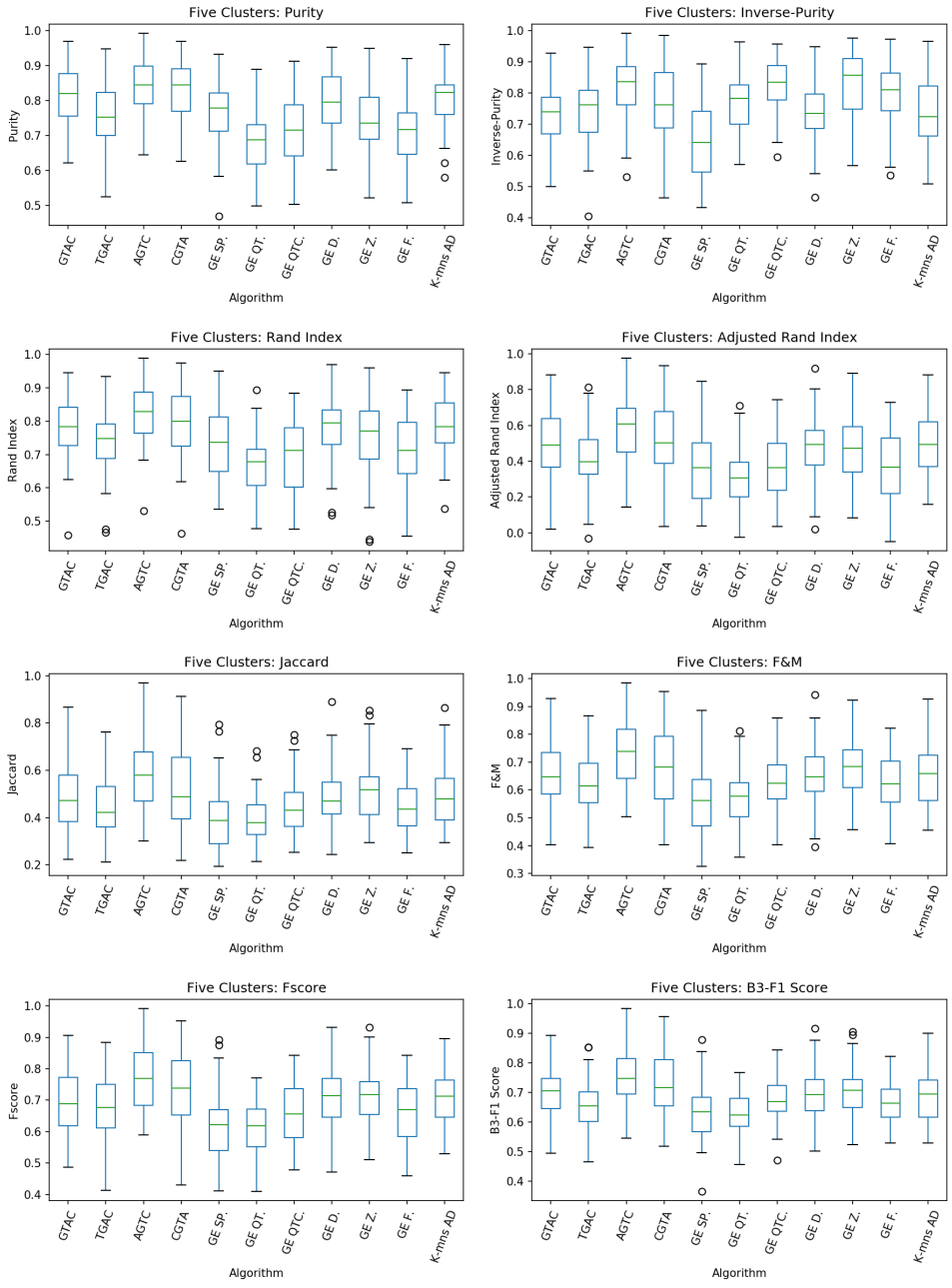

Fig. 17. Blox plot results for all tested configurations over the 51 web pages of the gold standard data set for all the external validation indices, for the task-oriented experiment.

Finally, in Table 8, we present the results of the Dunn statistical test to evaluate statistical differences between WPS configurations. Results show that AGTC and CATG configurations of the MCS algorithm are not statistically

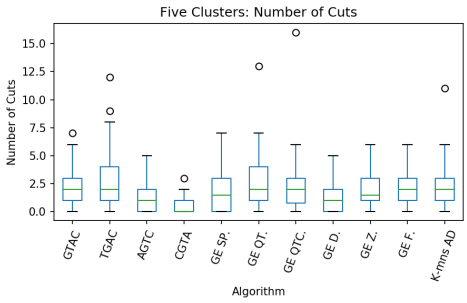

Fig. 18. Blox plot results for all tested configurations over the 51 web pages of the gold standard data set for internal validation index, number of cuts, for the task-oriented experiment.

different from each other (except for the number of cuts with some natural advantage for CATG), but they both show statistical differences with all other algorithms in terms F_{b^3}. This result must be attenuated for ARI, J, and F&M, as for ARI, differences are not observed with respect to K-means and GTAC, and for J and F&M, the same occurs with GE Z. Nevertheless, AGTC seems to perform superbly to all other configurations as this is the configuration with most statistical differences to all other tested algorithms. Indeed, CATG does not show statistical differences with many other configurations depending on the evaluation metric, confirmed by the large number of shared letters.

5.3 Analysis of Execution Time

Web page segmentation has received different attentions depending on its capacity to be run in real time or not. Within this context, most *ad hoc* solutions have been privileged for their capacity to segment web pages on the fly. Indeed, algorithms such as VIPS, BOM and BCS evidence fast execution times evaluated in terms of seconds for any given web page. Note that BCS shows most competitive results compared to VIPS, reducing the running time by a scale of up to 40 [87][48]. On the other hand, computer vision-based algorithms can be computationally expensive [41]. For example, the strategy proposed by [22] requires up to 1 hour for the larger web pages of their data set on a modern CPU. Likely, theoretically-founded strategies evidence high execution time, and cannot be executed in real time [37]. The same occurs for the MCS algorithm, which is not able to segment web pages on the fly. As mentioned in previous sections, this might not be an obstacle for some applications (namely within the context of the TagThunder project), although this issue must clearly be evaluated. As a consequence, we present the execution time statistics of most implemented algorithms in Table 9.

Algorithms	Median	Mean	Minimum	Maximum	Standard deviation
MCS	3658.2s	4896.9s	45.9s	19098.1s	4606.5s
GE SP.	141.8s	268.9s	20.1s	1489.1s	303.9s
GE QT.	930.2s	1922.1s	64.7s	9155.4s	2414.3s
GE QTC.	208.2s	1342.8s	20.1s	15723.9s	2785.9s
GE D.	31.6s	54.3s	4.2s	207.35	63.9s
K-means	70.5s	105.9s	0.1s	482.7s	125.2s

Table 9. Statistics of the execution time in seconds to segment 51 web pages for a variable number of clusters ($K = [3..8]$). Results are given for a server with a CPU with 12 cores (Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz).

[48] As such, results of BCS are better expressed in milliseconds.

Theoretically, the MCS algorithm shows a time complexity of $O(|N|^2 \times Kmax + |N| \times \mathbb{N}_b^2 + |N| \times Kmax^2)$, where $|N|$ is the number of chromosomes/assignments of the evolutionary process, $\mathbb{N}_b$ is the number of bboxes within a web page, and $Kmax$ is the upper bound of the number of clusters to be discovered. As such, it is clear that long web pages are particularly penalized in terms of execution time as illustrated in Table 9. Indeed, a very long page can take up to 5 hours when compared to smaller ones that can be executed in less than one minute. Note also that if the task at hand stands for the discovery of a large number of clusters, the MCS algorithm may take a long time to process.

However, the current implementation has not been optimized. For instance, it does not integrate multi-threading, which can "easily" be implemented. As such, our algorithm only uses one of the 12 cores of the CPU. As mentioned in section 2.4, different solutions can be studied to improve execution time, namely the ones proposed in [7, 33, 42, 58, 71], although this remains future work.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed to tackle Web Page Segmentation in a principled manner by integrating multimodal cues into a multi-objective K-means-based clustering algorithm, called MCS. MCS is parameter-free and does not depend on manually-tuned heuristics, but can only be run off-line. Comparatively to existing related works, MCS automatically finds the optimal number of clusters, combines into a single distance metric visual, logical and text semantics properties, and allows easy adaptation to different segmentation situations (variable or fixed number of clusters) due to its theoretical definition. Experimental results over an unconstrained WPS problem (variable number of clusters) clearly show that the ACXX and AGXX configurations of the MCS outperform all related works for a wide range of 8 external validation indices with statistical significance. The adaptation of MCS to a constrained WPS problem (fixed number of clusters, K=5) evidences similar results with the superiority of the AGTC configuration over all related works with statistical difference. In particular, related works included BCS [87], BOM [72] and GE series of algorithms [4], plus some straightforward implementations of the GE algorithm with the density-based QT algorithm [36]. Within this multi-criteria clustering situation, the visual properties clearly play an important role for WPS, luckily combined with logical properties within the unconstrained problem, being the textual cue the less discriminant feature.

Although conclusive results could be achieved within this set of experiments, a great deal of future work directions can be proposed. First, further experiments should be run on different data sets, eventually tackling different languages [40]. Second, as textual semantics features seem to be the less discriminant, more powerful models may be used, such as specifically-tuned transformer-based language models like BERT [25] or CamenBERT for the French language [54]. Text density features could also be introduced as proposed by [43] in the Block Fusion algorithm. Findings about text embedding maps [85] that combine visual and textual information into some latent space could also be an interesting research direction. Third, although current results show high performances for external validation indices, the number of discovered clusters is still much lower than the true situation. To overcome this issue, some extra objectives could be defined that try to determine some ideal clustering shapes (e.g. balanced vs. non balanced). Another solution resides in introducing an extra-parameter in current objectives so that large number of clusters would boost the objectives. Fourth, MCS can be tuned for other algorithms than K-means. As our problem is multi-criteria, this could be interesting to include multi-view versions of K-means as proposed in the following studies [19]. Following the same idea, this could be wise to weight each modality (visual, logical, textual) to better take into account their implication in the clustering process. Indeed, so far, each modality receives the exact same weight for the calculation of the distance between a bounding box and a given cluster. Another related research direction includes the implementation of multi-objective multi-criteria cluster ensemble techniques [62]. Finally, some extra studies should be performed to automatically

select the optimal solution from the Pareto-optimal front. An idea towards this direction lies in computing an extra meta-clustering step as proposed in [65], that would be able to find a consensus between all optimal solutions.

LIST OF ABBREVATIONS

ANC - Average number of clusters
AQMEA - Adaptive quantum-based multi-criterion evolutionary algorithm
ARI - Adjusted rand index
BF - Block fusion algorithm
BCS - Box clustering segmentation algorithm
BOM - Block-O-Matic algorithm
CPU - Central processor unit
CSS - Cascading style sheets
CUTS - Logical objective in terms of number of cuts
DB - Davies-Bouldin index
DBV - Davies-Bouldin index for border-to-border distance
DBT - Davies-Bouldin index for text dissimilarity
DoC Degree of coherence
DOM - Document object model
FM - Folks and mallows
GE - Guided expansion algorithm
GE SP. - Guided expansion algorithm with simple pre-process
GE QT. - Guided expansion algorithm with QT pre-processing
GE QTC. - Guided expansion algorithm with complete QT pre-processing
HEPS - Heading-based page segmentation algorithm
INP - Inverse purity
J - Jaccard coefficient
QMEA - Quantum-inspired multi-objective evolutionary algorithm
QT - Quality threshold clustering
MCS - Multi-objective clustering segmentation algorithm
NSGA-II - Non-dominated sorting genetic algorithm
NMI - Normalized mutual information
P - Purity
RI - Rand index
SIA - Silhouette index for alignment
SOM - Self-organizing maps
URL - Uniform Resource Locator
VIPS - Vision-based page segmentation algorithm
WebSeg - Web segmentation tool
WPS - Web page segmentation

REFERENCES

[1] Sadet Alcic and Stefan Conrad. 2011. Page Segmentation by web content clustering. In *International Conference on Web Intelligence, Mining and Semantics (WIMS)*. 1–9.

[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.

[3] J-J. Andrew. 2020. *Task Oriented Web Page Segmentation.* Ph.D. Dissertation. University of Caen Lower Normandy.

[4] J-J. Andrew, S. Ferrari, F. Maurel, G. Dias, and E. Giguet. 2019. Model-driven web page segmentation for non visual access. In *16th International Conference of the Pacific Association for Computational Linguistics (PACLING)*.

[5] J-J. Andrew, S. Ferrari, F. Maurel, G. Dias, and E. Giguet. 2019. Web Page Segmentation for Non Visual Skimming. In *33rd Pacific Asia Conference on Language, Information and Computation (PACLIC)*.

[6] David Arthur and Sergei Vassilvitskii. 2007. K-Means++: The advantages of careful seeding. *18th Annual ACM Symposium on Discrete Algorithms (SIAM)* 8, 1027–1035.

[7] Jerzy Balicki. 2009. An Adaptive Quantum-Based Multiobjective Evolutionary Algorithm for Efficient Task Assignment in Distributed Systems. In *13th WSEAES International Conference on Computers (ICCOMP)*. 417–422.

[8] Shumeet Baluja. 2006. Browsing on small screens: Recasting web-page segmentation into an efficient machine learning framework. In *15th International Conference on World Wide Web (WWW)*. 33–42.

[9] 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 *37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*. 767–776.

[10] Adelbert W Bronkhorst. 2000. The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica* 86, 1 (2000), 117–128.

[11] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. 2004. Hierarchical clustering of WWW image search results using visual, textual and link information. In *12th Annual ACM International Conference on Multimedia (MM)*. 952–959.

[12] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Extracting content structure for web pages based on visual representation. In *5th Asia-Pacific Web Conference on Web Technologies and Applications*. 406–417.

[13] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-view K-means clustering on big data. In *23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2598–2604.

[14] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR* abs/1803.11175 (2018).

[15] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. 2008. A graph-theoretic approach to webpage segmentation. In *17th International Conference on World Wide Web (WWW)*. 377–386.

[16] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. 2019. Hybrid Task Cascade for Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 4969–4978.

[17] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. arXiv:1906.07155 [cs.CV]

[18] Yu Chen, Wei-Ying Ma, and Hong-Jiang Zhang. 2003. Detecting web page structure for adaptive viewing on small form factor devices. In *12th International Conference on World Wide Web (WWW)*. 225–233.

[19] G. Cleuziou, M. Exbrayat, L. Martin, and J. Sublemontier. 2009. CoFKM: A centralized method for multiple-view clustering. In *9th IEEE International Conference on Data Mining (ICDM)*. 752–757.

[20] S. Coondu, S. Chattopadhyay, M. Chattopadhyay, and S. R. Chowdhury. 2014. Mobile-enabled content adaptation system for e-learning websites using segmentation algorithm. In *8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. 1–8.

[21] Courtney D Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*. 13–18.

[22] Michael Cormer, 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)*. 345–352.

[23] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 (1979), 224–227.

[24] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805 (2018).

[26] Richard Dubes and Anil K. Jain. 1980. Clustering methodologies in exploratory data analysis. In *Advances in computers*. Vol. 19. Elsevier, 113–228.

[27] Olive Jean Dunn. 1964. Multiple Comparisons Using Rank Sums. *Technometrics* 6, 3 (1964), 241–252.

[28] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *2nd International Conference on Knowledge Discovery and Data Mining (KDD)*. 226–231.

[29] Vladimir Estivill-Castro. 2002. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter* 4, 1 (2002), 65–75.

[30] Maurel F., Dias G., Ferrari S., Andrew J-J., and Giguet E. 2019. Concurrent Speech Synthesis to Improve Document First Glance for the Blind. In *2nd International Workshop on Human-Document Interaction (HDI) associated to 15th International Conference on Document Analysis (ICDAR)*. 10–17.

[31] AM Fahim, AM Salem, F Af Torkey, and MA Ramadan. 2006. An efficient enhanced k-means clustering algorithm. *Journal of Zhejiang University-Science A* 7, 10 (2006), 1626–1633.

[32] Stéphanie Giraud, Pierre Thérouanne, and Dirk D Steiner. 2018. Web accessibility: Filtering redundant and irrelevant information improves website usability for blind users. *International Journal of Human-Computer Studies* 111 (2018), 23 – 35.

[33] Yue-Jiao Gong, Wei-Neng Chen, Zhi-Hui Zhan, Jun Zhang, Yun Li, Qingfu Zhang, and Jing-Jing Li. 2015. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing* 34 (2015), 286–300.

[34] João Guerreiro. 2015. The Use of Concurrent Speech to Enhance Blind People's Scanning for Relevant Information. *SIGACCESS Accessibility and Computing* 111 (2015), 42–46.

[35] Julia Handl and Joshua Knowles. 2007. An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation* 11, 1 (2007), 56–76.

[36] Laurie J Heyer, Semyon Kruglyak, and Shibu Yooseph. 1999. Exploring expression data: identification and analysis of coexpressed genes. *Genome research* 9, 11 (1999), 1106–1115.

[37] Zexun Jiang, Hao Yin, Yulei Wu, Yongqiang Lyu, Geyong Min, and Xu Zhang. 2019. Constructing novel block layouts for webpage analysis. *ACM Transactions on Internet Technology* 19, 3 (2019), 1–18.

[38] Lecarpentier J.M., Manishina E., Maurel F., Ferrari S., Giguet E., Dias G., and Busson M. 2016. Tag Thunder: Web Page Skimming in Non Visual Environment Using Concurrent Speech. In *7th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT) associated to INTERSPEECH*. 1–8.

[39] Yunjae Jung, Haesun Park, Ding-Zhu Du, and Barry L Drake. 2003. A decision criterion for the optimal number of clusters in hierarchical clustering. *Journal of Global Optimization* 25, 1 (2003), 91–111.

[40] Johannes Kiesel, Florian Kneist, Lars Meyer, Kristof Komlossy, Benno Stein, and Martin Potthast. 2020. Web Page Segmentation Revisited: Evaluation Framework and Dataset. In *29th ACM International Conference on Information Knowledge Management (CIKM)*. 3047–3054.

[41] Johannes Kiesel, Lars Meyer, Florian Kneist, Benno Stein, and Martin Potthast. 2021. An Empirical Comparison of Web Page Segmentation Algorithms. In *43rd European Conference on IR Research (ECIR)*.

[42] Yehoon Kim, Jong-Hwan Kim, and Kuk-Hyun Han. 2006. Quantum-inspired Multiobjective Evolutionary Algorithm for Multiobjective 0/1 Knapsack Problems. In *IEEE International Conference on Evolutionary Computation (ICEC)*. 2601–2606.

[43] Christian Kohlschütter and Wolfgang Nejdl. 2008. A densitometric approach to web page segmentation. In *17th ACM Conference on Information and Knowledge Management (CIKM)*. 1173–1182.

[44] Teuvo Kohonen. 1982. Self-organized formation of topologically correct feature maps. *Biological cybernetics* 43, 1 (1982), 59–69.

[45] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. 2011. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, 3 (2011), 231–240.

[46] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *31st International Conference on Machine Learning (ICML)*. 1188–1196.

[47] Alison Lee and Vicki Hanson. 2003. Enhancing web accessibility. In *11th Annual ACM International Conference on Multimedia (MM)*. 456–457.

[48] Aristidis Likas, Nikos Vlassis, and Jakob [J. Verbeek]. 2003. The global k-means clustering algorithm. *Pattern Recognition* 36, 2 (2003), 451 – 461.

[49] 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 *13th European Conference on Computer Vision (ECCV)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). 740–755.

[50] S. Lloyd. 2006. Least Squares Quantization in PCM. *IEEE Transactions of Information Theory* 28, 2 (2006), 129–137.

[51] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. 281–297.

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

[53] Elena Manishina, Jean-Marc Lecarpentier, Fabrice Maurel, Stéphane Ferrari, and Maxence Busson. 2016. Tag thunder: Towards non-visual web page skimming. In *18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*. 281–282.

[54] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 7203–7219.

[55] Fabrice Maurel. 2004. *Transmodalité et multimodalité écrit/oral : modélisation, traitement automatique et évaluation de stratégies de présentation des structures "visuo-architecturale" des textes.* Ph.D. Dissertation. Université de Toulouse.

[56] Benjamin Meier, Thilo Stadelmann, Jan Stampfli, Marek Arnold, and Mark Cieliebak. 2017. Fully Convolutional Neural Networks for Newspaper Article Segmentation. In *14th International Conference on Document Analysis and Recognition (ICDAR)*. 414–419.

[57] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance M. Kaplan, and Jiawei Han. 2019. Spherical text embedding. In *32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*. 8206–8215.

[58] Souham Meshoul, Karima Mahdi, and Mohamed Batouche. 2005. A Quantum Inspired Evolutionary Framework for Multi-objective Optimization. In *Progress in Artificial Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 190–201.

[59] Martin Milicka and Radek Burget. 2015. Information extraction from web sources based on multi-aspect content analysis. In *Semantic Web Evaluation Challenges*. Springer, 81–92.

[60] George A Miller. 1994. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 101, 2 (1994), 343.

[61] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[62] Sayantan Mitra and Sriparna Saha. 2019. A multiobjective multi-view cluster ensemble technique: Application in patient subclassification. *PLOS ONE* 14, 5 (05 2019), 1–30.

[63] Jose G. Moreno and Gaël Dias. 2015. Adapted B-CUBED metrics to unbalanced datasets. In *38th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 911–914.

[64] Jose G. Moreno, Gaël Dias, and Guillaume Cleuziou. 2014. Query log driven web search results clustering. In *37th International ACM Conference on Research & Development in Information Retrieval (SIGIR)*. 777–786.

[65] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay. 2009. Multiobjective genetic clustering with ensemble among pareto front solutions: Application to MRI brain image segmentation. In *7th International Conference on Advances in Pattern Recognition (ICPRAM)*. 236–239.

[66] Anirban Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. 2015. A survey of multiobjective evolutionary clustering. *ACM Computing Survey* 47, 4 (2015).

[67] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65.

[68] Waseem Safi, Fabrice Maurel, Jean-Marc Routoure, Pierre Beust, and Gaël Dias. 2015. Web-Adapted Supervised Segmentation to Improve a New Tactile Vision Sensory Substitution (TVSS) Technology. *Procedia Computer Science* 52 (2015), 35–42. 6th International Conference on Ambient Systems, Networks and Technologies (ANT).

[69] Sriparna Saha and Sanghamitra Bandyopadhyay. 2010. A symmetry based multiobjective clustering technique for automatic evolution of clusters. *Pattern recognition* 43, 3 (2010), 738–751.

[70] Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2018. Automatic scientific document clustering using self-organized multi-objective differential evolution. *Cognitive Computation* (12 2018), 1–23.

[71] Pasquale Salza and Filomena Ferrucci. 2019. Speed up genetic algorithms in the cloud using software containers. *Future Generation Computer Systems* 92 (2019), 276–289.

[72] Andrés Sanoja and Stéphane Gançarski. 2014. Block-o-Matic: A web page segmentation framework. In *International Conference on Multimedia Computing and Systems (ICMCS)*. 595–600.

[73] Andrés Sanoja and Stéphane Gançarski. 2015. Web page segmentation evaluation. In *30th Annual ACM Symposium on Applied Computing (SAC)*.

[74] Andrés Sanoja Vargas. 2015. *Web page segmentation, evaluation and applications*. Ph.D. Dissertation. Pierre and Marie Curie University, Paris, France.

[75] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Communications of ACM* 63, 12 (2020), 54–63.

[76] Xavier Sevillano, Joan Claudi Socoró, and Francesc Alías. 2020. Parallel hierarchical architectures for efficient consensus clustering on big multimedia cluster ensembles. *Information Sciences* 511 (2020), 212 – 228.

[77] Roberto Panerai Velloso and Carina F. Dorneles. 2017. Extracting records from the web using a signal processing approach. In *2017 ACM on Conference on Information and Knowledge Management (CIKM)*. 197–206.

[78] Roberto Panerai Velloso and Carina F. Dorneles. 2019. Web page structured content detection using supervised machine learning. In *International Conference on Web Engineering (ICWE)*, Maxim Bakaev, Flavius Frasincar, and In-Young Ko (Eds.). 3–18.

[79] Daiyue Weng, Jun Hong, and David A. Bell. 2011. Extracting data records from query result pages based on visual features. *Advances in Databases* (2011), 140–153.

[80] Daiyue Weng, Jun Hong, and David A. Bell. 2014. Automatically annotating structured web data using a SVM-based multiclass classifier. In *15th International Conference on Web Information Systems Engineering (WISE)*, Boualem Benatallah, Azer Bestavros, Yannis Manolopoulos, Athena Vakali, and Yanchun Zhang (Eds.). 115–124.

[81] Lucas Wiener, Tomas Ekholm, and Philipp Haller. 2017. Modular Responsive Web Design: An Experience Report. In *Companion to the First International Conference on the Art, Science and Engineering of Programming*. Association for Computing Machinery, Article 22, 6 pages.

[82] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-Training of Text and Layout for Document Image Understanding. In *26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD)*. 1192–1200.

[83] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2020. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. arXiv:2012.14740 [cs.CL]

[84] Xin Yang and Yuanchun Shi. 2007. Web page segmentation based on gestalt theory. In *IEEE International Conference on Multimedia and Expo (ICME)*. 2253–2256.

[85] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5315–5324.

[86] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual Universal Sentence Encoder for Semantic Retrieval. arXiv:1907.04307

[87] Jan Zeleny, Radek Burget, and Jaroslav Zendulka. 2017. Box clustering segmentation: A new method for vision-based web page preprocessing. *Information Processing & Management* 53, 3 (2017), 735–750.

[88] Shibing Zhou, Zhenyuan Xu, and Fei Liu. 2016. Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. *IEEE Transactions on Neural Networks and learning systems* 28, 12 (2016), 3007–3017.