# Extracting Informative Textual Parts from Web Pages Containing User-Generated Content

Nikolaos Pappas*
Idiap Research Institute
Rue Marconi 19
Martigny 1920, Switzerland
nikolaos.pappas@idiap.ch

Georgios Katsimpras
University of the Aegean
Department of Information and Communication Systems Engineering
Karlovassi 83200, Greece
gkatsimpras@gmail.com

Efstathios Stamatatos
University of the Aegean
Department of Information and Communication Systems Engineering
Karlovassi 83200, Greece
stamatatos@aegean.gr

## ABSTRACT

The vast amount of user-generated content on the Web has increased the need for handling the problem of automatically processing content in web pages. The segmentation of web pages and noise (non-informative segment) removal are important pre-processing steps in a variety of applications such as sentiment analysis, text summarization and information retrieval. Currently, these two tasks tend to be handled separately or are handled together without emphasizing the diversity of the web corpora and the web page type detection. We present a unified approach that is able to provide robust identification of informative textual parts in web pages along with accurate type detection. The proposed algorithm takes into account visual and non-visual characteristics of a web page and is able to remove noisy parts from three major categories of pages which contain user-generated content (News, Blogs, Discussions). Based on a human annotated corpus consisting of diverse topics, domains and templates, we demonstrate the learning abilities of our algorithm, we examine its effectiveness in extracting the informative textual parts and its usage as a rule-based classifier for web page type detection in a realistic web setting.

## Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval — Information filtering; I.2.6 [**Artificial Intelligence**]: Learning — Parameter learning

## General Terms

Algorithms, Design, Performance, Experimentation

## Keywords

Web Page Segmentation, Noise Removal, Information Extraction, Web Page Type Detection

*Research performed partly while at University of the Aegean and partly while at Idiap Research Institute.


*i-Know'12,* September 05-07, 2012, Graz, Austria


## 1. INTRODUCTION

A huge amount of user-generated content is being published every day in social networks, news, blogs and other sources in the Web. Many web-driven applications such as: brand analysis, measuring marketing effectiveness, influence networks, customer experience management and many other, are utilizing this content in order to extract and process valuable information. While Web 2.0 democratized content publishing by users, extracting and processing the published content still remains a difficult task. The extraction of the valuable information from the web pages can be performed by the usage of application specific APIs, which give machine-readable access to the contents directly from the source. Naturally, this simple approach is impractical when there are API usage limits or, more importantly, when many different sources have to be examined.

Most applications dealing with Web content require two basic pre-processing steps: web page segmentation and noise removal. The former can divide a page into multiple semantically coherent parts as in [3, 7, 17, 11, 1]. The latter filters out the parts that are not considered useful (e.g. ads, banners, etc.) as in [19, 21, 4, 10, 15, 16]. In the majority of the applications these pre-processing tasks are handled separately. Those studies that are addressing them together, are evaluated using web pages from specific sites (like in [19, 21, 16, 4]), on different context (like in [10]), or by using limited number of specific blog feeds (like in [15]). The mentioned approaches do not emphasize the diversity of the examined dataset or it is assumed that two or more pages from the same domain will always be available (labeled) in order to apply template detection and removal but this is not always true (e.g. during a web crawling task). Moreover there are regions in web pages that change dynamically (e.g related items with description) but still can be considered as noise, a template-based noise removal method would fail to detect that. To our knowledge there is no previous work that has been evaluated based on such a diverse dataset using a metric that penalizes the algorithm with respect to the exact regions captured and combines informative textual part extraction along with web page type detection.

In this paper, we propose a unified approach to handle both of these pre-processing steps by focusing on the extraction of informative textual parts. The proposed method is based on both visual and non-visual features of web pages and it is able to handle three major types of web pages usually comprising user-generated content: articles (as in

online newspapers, magazines), articles with comments (as in blogs, news), and discussion pages (as in forums, review sites, question asking sites). The categorization of the page types refers to the semantical interpretation of different web pages and not to specific web page templates, thus a web page type based on our model may include different representational templates. Moreover, unlike other work, the proposed method does not rely on consecutive similar pages for finding noise patterns ([15]) or for detecting the page template ([16]). Instead it relies on local features of each page and two statistical thresholds that can be learned on a small set of samples of a given population.

The evaluation of the method was performed using a human annotated corpus obtained from a general web search engine (Google) and the pages were sampled from News, Blogs, Discussions categories. The categories selected for the corpus contain a variety of different topics, domains and templates, thus we argue that our experimental results are significant and can be generalized to the population of web pages on these categories. The estimation of the optimal thresholds was performed on a small subset of this corpus (20%) and evaluated on the rest obtaining promising results (about 80% average accuracy on each category separately and 78% on all categories together).

Moreover, we demonstrate the ability of the method to handle unknown instances of web pages i.e. evaluation without knowing a priori the type of the web page examined. Despite the extraction of the semantic regions in our approach, the final output contains also the relations between them, giving additive value to the output. Using this information we can make a grouping of the web pages that share similar features i.e. identify the type of the examined page. Therefore, we performed a set of experiments using the proposed algorithm as a rule-based classifier in various settings to evaluate its coverage and performance on the task of identification of the three major types of web pages.

The proposed method is easy to follow and provides a robust solution in the extraction of informative noise-free textual parts from web pages. The next section discusses previous work while Section 3 describes our approach in detail. Section 4 includes the experiments and Section 5 summarizes the conclusions drawn from this study.

## 2. RELATED WORK

The approaches found in the literature mostly detect and semantically annotate the segments (blocks) of the page and fewer studies are dealing with the problem of removing noisy (non-informative) segments. The less sophisticated methods for web page segmentation rely on building wrappers for a specific type of web pages. Some of these approaches rely on hand crafted web scrapers that use hand-coded rules specific for certain template types [13]. The disadvantage of this approach is that they are very inflexible and unable to handle the template changes in web pages.

The methods applied for the solution of the web page segmentation problem are using a combination of non-visual and (or) visual characteristics. Examples of non-visual based methods are presented by Diao [8] who treats segments of web pages in a learning based web query processing system and deals with major types of HTML tags (<p>, <table>, etc.). Lin [14] only considers the table tag and its offspring as a content block, uses an entropy based approach to discover informative ones. Gibson et al. [9] considers element frequencies for template detection while Debnath et al. [7] compute an inverse block frequency for classification. In [5], Chakrabarti et al. determine the "templateness" of DOM[1] nodes by regularized isotonic regression. Yi et al. [19] simplify the DOM structure by deriving a so-called Site Style Tree which is then used for classification. Vineel proposed a DOM tree mining approach based on Content Size and Entropy which is able to detect repetitive patterns [17]. Kang et al. also proposed a Repetition-based approach for finding patterns in the DOM tree structure [11]. Alcic et al. investigate the problem from a clustering point of view by using distance measures for content units based on their DOM, geometric and semantic properties [1].

A well-known method based on visual characteristics is VIPS by Cai et al. [3] which is an automatic top-down, tag-tree independent approach to detect web content structure. It simulates how a user understands web layout structure based on his visual perception. Another example is the approach by Chen et al. to tag pattern recognition [6] as well as Baluja's [2] method using decision tree learning and entropy reduction. HuYan and MiaoMiao [18] proposed a multi-cue algorithm which uses various information: visual information (background color, font size), some non-visual information (tags), text information and link information. Cao et al. used vision and effective text information to locate the main text of a blog page and the information quantity of separator to detect the comments [4]. A. Zhang et al. [20] focused on precise web page segmentation based on semantic block headers detection using visual and structural features of the pages. Finally, Kohlschütter et al. proposed an approach by building on methods from quantitative linguistics and computer vision [12].

Concerning the noise removal problem, Yi et al. [19] observed that noisy blocks share some common content and presentation style, and the main blocks of pages diverse. A style-based tree was used with an information based measure which evaluates the importance of each node. J. Mayfield [10] focused mainly on noise removal of blog data based on local features of link tags in a page applied for spam blog detection and improvement of Retrieval task. K. Vieira et al. [16] proposed a template removal method for web pages that uses a small set sample of pages per site for the template detection. D. Cao et al. S. Nam et al. [15] perform filtering of non-relevant content on a page based on content difference between two consequent blog posts in the same blog site. [4] combined vision and effective text information to locate main text and comments in blog pages. Zhang and Deng [21] establish a block tree model by combining DOM tree and visual characteristics of web content and a statistical learning method using neural networks.

## 3. THE PROPOSED METHOD

As mentioned earlier, our method processes a web page and segments it into semantic parts while it ignores noisy content. The algorithm used in our approach (called SD Algorithm), exploits visual and non-visual characteristics of a web page encapsulated in a DOM tree with additional features called SD-Tree and performs the page type classification and region extraction using the optimal values of

[1]Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents.

its statistical thresholds obtained from a small subset of the given corpus (see Section 4). Below, we describe in detail the characteristics of our method (namely the features), the SD-Tree structure that is being used by the SD Algorithm and finally the steps of the SD Algorithm along with an example.

## 3.1 Characteristics

The method makes use of a combination of non-visual and visual characteristics of a web page in order to achieve the page segmentation and the filtering of noisy areas. In the following list we present all these characteristics that were used along with a detailed description.

- **DOM Structure**
  In order to handle a structured document written in HTML or XML, more efficiently and consistently, the World Wide Web Consortium (W3C) published the Document Object Model (DOM) specification. DOM gives the ability to access and manipulate information stored in a structured HTML or XML document.

- **HTML element tags**
  HTML DOM is in a tree structure, usually called an HTML DOM tree. In general, every node in the DOM tree can represent a visual block. However, some nodes such as <table> and  are used only for organization purpose and are not appropriate to represent a single visual block. Therefore, we use the following categorization of nodes that are used by the algorithm in order to proceed to valid merging of nodes.

  - **Strong tags**: A type of node can be used to divide the structure of a web page or organize the content of a web page, such as <table>, <tr>, , <ui>, <tbody>, etc.
  - **Weak tags**: This type of node is simply to display the contents of web page and usually included in the organized node as an internal node such as <td>, <li>, <p>, <img>, etc.

- **Density**
  The density is used to define the size of a node and is represented by the number of characters that are contained in a specific HTML node. It is very useful for comparing nodes based on their size.

- **Distance from max density region**
  It describes how "far" a region is from another with respect to a density feature. The distance from the max density region is calculated by the following formula:

$$dfm(r) = 100 - (d_r * 100)/d_{max} \quad (1)$$

  where $d_r$ is the density of the examined region $r$ and $d_{max}$ is the density of the region with the max density. A high value of *dfm* for a specific region $r$ means that we have to deal with a small region in the document, in contrast, small values of *dfm* represent bigger regions.

- **Distance from root**
  It describes how "far" a region is from the root node of the DOM tree. It is simply calculated by the number of parents of a node until the root node i.e. the level (or depth) of the tree in which the node is entered.

- **Ancestor title**
  The ancestor title is the title detected in some of the ancestor nodes of a specific node. The title is expressed by the <h1>, <h2>, <h3>, etc. HTML tags. An area that has an ancestor title in a close level in the DOM tree could mean that this region is important with respect to its content. Usually, the programmers highlight the content of this specific region by specifying a title (e.g. a title for an article, title for side regions etc.). The above properties give semantic value to this non-visual characteristic.

- **Ancestor title level**
  The DOM level in which the ancestor title of a node was detected. With this metric we can calculate the level by which the ancestor title differs from the node level in the DOM tree. It defines how "far" an ancestor title is from a specific node with respect to the tree level.

$$title\_diff_{node} = |level_{anc\_title} - level_{node}| \quad (2)$$

  where $level_{anc\_title}$ is the level of an ancestor title and $level_{node}$ is the level of the examined node. A high value for $title\_diff$ for a specific *node* means that the title has less possibilities to refer to a node's content, in contrast small values increases the possibility.

- **Cardinality**
  The cardinality of a node is simply the number of elements that a node contains, i.e. the number of child nodes.

- **Content of HTML nodes**
  The content of the HTML nodes is the text that they contain. It is used mainly when we want to scan for keywords matching the comments context i.e. to detect comment regions.

- **CSS classes and individual styles. (Visual)**
  Finally, unlike the previous non-visual characteristics, we take also into account visual information from CSS classes and styles. Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML. The CSS classes can be defined in the HTML attribute class of each element. Usually, the elements that share common visual representation belong to the same CSS class. The individual styles contain additional visual information on the element e.g. the visibility or the size of an element.

## 3.2 SD-Tree

The SD-Tree (Style-Density Tree) is an HTML DOM tree with features concerning the style and the density of a node. Each node is described by a variety of features explained earlier: Parent node, Child nodes, Tag, Cardinality, Text, Distance from root (*dfr*), Distance from max density region (*dfm*), Class, Id, Style, Ancestor title and Ancestor title level. For the construction of the SD-Tree, initially the creation of the HTML DOM tree is performed and right after the additional features for each of the nodes are calculated.

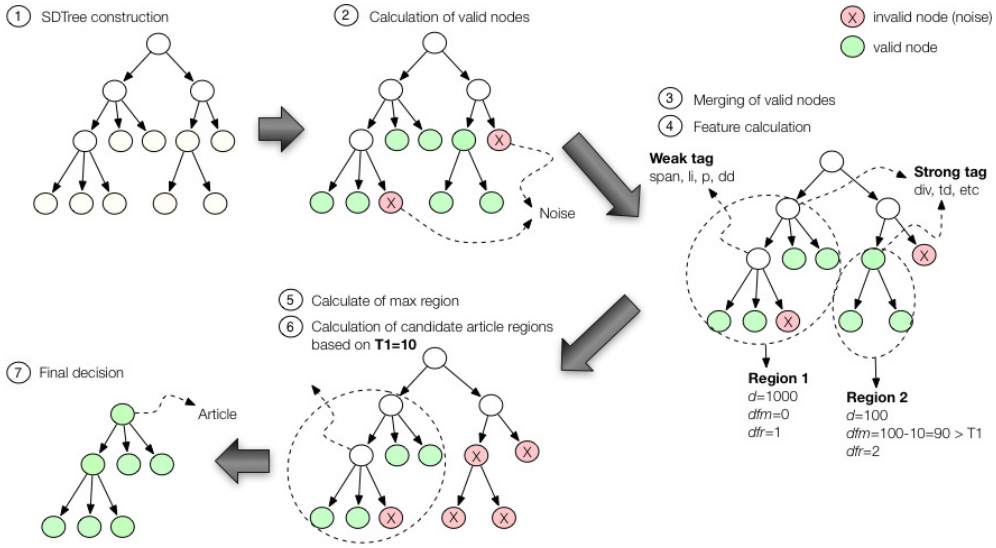


**Figure 1: Example of SD Algorithm following steps 1 to 7 and resulting in an Article.**

The SD-Tree is created at the initial stage of the SD Algorithm and is then used further for the calculation of the semantic regions.

## 3.3 SD Algorithm

The SD Algorithm recognizes the type of pages (Article, Comments, Multiple areas) and extracts their constituent regions. The algorithm ignores the noisy areas (the side panels, footer, header etc.) and keeps only the regions that have meaningful content. Two thresholds are used, the max density region distance threshold **T1** and the min region density threshold **T2**. The **T1** threshold, defines the max allowed distance from max density region, so for each node the threshold is compared against the distance from max density region (*dfm*). The **T2** threshold, defines the min allowed density for a node in order not to be treated as noise. The algorithm, using **T1** and **T2** proceeds as follows:

1. **Construction of SD-Tree**
   In this step the construction of the SD-Tree is performed based on the HTML content of the web page. At this initial construction only the density feature and DOM features are calculated (e.g. tags, parents, children, etc.).

2. **Valid nodes calculation based on T2 (Pruning)**
   The valid nodes are all the nodes that have density greater than **T2** threshold. In this step all the valid nodes are calculated and the nodes containing noise are ignored. At this stage also the nodes that have attributes in their CSS styles that imply invisibility on the page (e.g. display:none, visibility:hidden), are completely ignored.

3. **Merging of valid nodes into valid groups**
   All the valid nodes are merged into bigger valid groups called regions if it is possible. For each node that has a tag which belongs to weak tags, it is merged to its parent in case it represents a strong tag. If the parent has a weak tag, then the parent of the parent is checked. The process repeats until a parent with a strong tag is found.

4. **Feature calculation for valid groups (density, ancestor title, cardinality, etc.)**
   After the calculation of the valid groups, the features for each of those nodes are calculated. We calculate all the additional features in this stage in order to avoid some useless calculations (e.g. for some nodes that will not be used in further steps). In the first steps (1-3) of the algorithm only the density and the tag features are needed.

5. **Calculation of max density region and distances from max density region**
   The regions are compared based on their density and the max region density is found. Then the distances from this max region (*dfm*) feature is calculated for all regions.

6. **Candidate article regions detection based on T1**
   At this step all the regions that have distance from max (*dfm*) less or equal to threshold **T1** are the candidate article regions and all regions are grouped based on their CSS classes.

7. **Make final decision**
   The final decision is made based on the candidate article regions detected from the previous steps.

   - **Calculate article region**
     The region among the candidates, closer to the root node and with an ancestor title to the closest level is denoted as article region.
   - **if** *article_region* **found**
     When the candidates are greater than zero and an article was detected, further examination is made to the candidates in order to detect the page type.
   - **Scan for comment regions in the candidates**
     All the remaining regions are examined whether they are comment regions or not. At this step the Content (Text), the CSS classes and Id features of candidate regions are scanned for keywords that

belong to the comment keywords specification[2]. Also it is checked whether or not these regions have a common parent with Article region.

  - **if** *comment_regions* $> 0$ **return Article with Comments**
  - **else if (all** *candidate_regions* **in same level) return Multiple**
  - **else return Article**
- **else return Multiple**

## 3.4 Example output of SD Algorithm

To illustrate the proposed algorithm, we applied it to a page that contains a single article (Fig. 1). Initially, the construction of the SD-Tree is made and the valid nodes are calculated. At this step some nodes are considered as noise. At the next step the merging of the valid nodes is made and the regions **R1**, **R2** are formed and the features of each of these regions are calculated. The next step is the calculation of the max region, which is the **R1** with density $d$ 1000. Then the calculation of the candidate article regions is made by ignoring all the regions that have distance from max density region ($dfm$) greater than threshold T1=10. Therefore, the region **R2** is ignored and the final decision is taken with result the **R1** region as an Article.

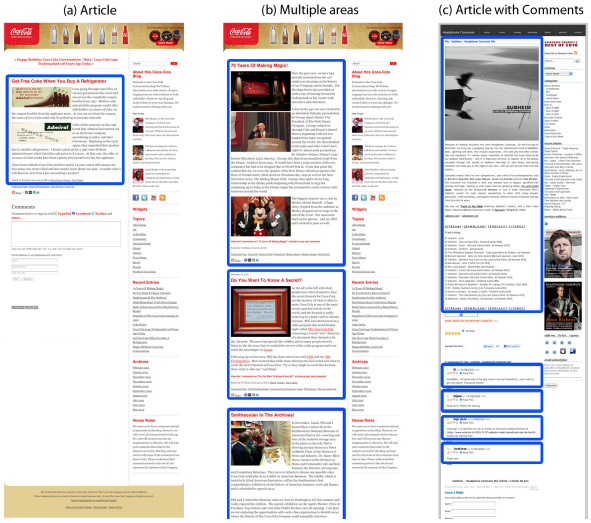


**Figure 2: Example output of SD Algorithm for different web page types (solid square outline): (a) Articles, (b) Multiple areas, (c) Articles with Comments. In all three cases the precise areas of interest have been captured.**

In Fig. 2 we demonstrate an example output of the SD Algorithm for different page types. In Fig. 2 (a) we can see an example of Article region where only the specific area is extracted from the algorithm. The same for Fig. 2 (b), all the multiple areas were only extracted and the other regions were treated as noise. Finally the Fig. 2 (c) contains the article and the comments that were written underneath. Intuitively, the extracted structural information can be exploited for topic-based opinion retrieval and mining.

[2]A set of keywords that we have defined in order to detect comment regions: COMMENT_TAGS = ['comment', 'reply', 'response', 'user', 'wrote:', 'said:'].

# 4. EXPERIMENTAL SETUP

In this section we analyze the experimental setup concerning the evaluation of the SD Algorithm. We provide analytical details about the datasets and the evaluation metrics that were used. Moreover, we perform a diversity comparison of the datasets used in previous work with the datasets used in our evaluation.

## 4.1 Datasets

For the evaluation of the SD Algorithm we used two annotated corpora called SD Diverse and SD Non-Diverse, that contain the three different web page types (or classes) that are examined (Articles, Articles with Comments and Multiple areas). Their difference is on the variety on the domains, templates and number of web pages. In addition, the former was produced by human annotation while the latter was built automatically.

### *4.1.1 SD Diverse Dataset*

The first one called SD Diverse is a human-annotated corpus comprising of 600 web pages acquired from News, Blogs, Discussions categories with the Google web search engine. It contains a great variety of different topics (politics, arts, sports, products, music, etc), domains (that consist of different templates); namely 521 distinct different domains. This corpus contains the following human annotated classes and annotated informative regions:

- **Articles (200 pages)**
  Consists of pages that contain a distinct article region without any user comments related to this region.
- **Articles with Comments (200 pages)**
  Consists of pages that contain an article along with a number of comments related to this article region.
- **Multiple areas (200 pages)**
  Consists of pages that contain multiple similar regions (e.g. forum discussions, multiple articles in a blog or news, answers etc.).

### *4.1.2 SD Non-Diverse Dataset*

The second dataset called SD Non-Diverse consists of 5400 web pages belonging to 10 domains (randomly selected from the previous dataset) and it was automatically annotated using domain-specific heuristics. Similar to the previous dataset it comprises of three classes: Articles (1800 pages), Articles with Comments (1800 pages) and Multiple areas (1800 pages). The annotation of the semantic regions was done automatically as well based on human annotations for each combination of domain with the three classes.

## 4.2 Diversity comparison

The comparison between different methods addressing noise removal is a difficult task since each of the previous work is evaluated in different datasets and in different terms. Moreover, some of the methods do not emphasize the web segmentation problem i.e. extracting and annotating informative regions precisely, instead they evaluate some other task (e.g. information retrieval). We believe that since there is no available annotated dataset for this specific task (informative regions), the evaluation has to be done with more strict criteria and in terms of the diversity of the dataset examined. The variety of a dataset is crucial to estimate

the effectiveness of the examined algorithm on various tasks (information retrieval, page classification and clustering) but most importantly we can produce very precise informative content that is very useful and critical in natural language processing tasks (sentiment analysis, text summarization).

The reason why we emphasize the diversity of the dataset is based on the observation that pages in same domain or specific blog feeds share the same template. In fact some methods have relied on this observation and were evaluated in terms of removing noise from a lot of pages belonging to a small amount of different domains. We should note here that some sites like blogspot.com include other domains with different templates but none of the examined previous work used such sites for their evaluation (they only used limited underlying domains that belong to such sites). As we mentioned earlier, pages from the same domain are not always available (e.g. for web crawling, or topic-based crawling). Consequently, a method has to be evaluated based on its performance on a large amount of different domains and templates and not on a huge amount of pages from specific domains, otherwise the good performance that they may have in limited domains will not ensure its significance in large scale.

In order to quantify the above observations we define the notion of *Diversity* of a dataset $C$ using simply the ratio of the number of distinct domains $n_{domains \in C}$ that consist of different templates, to the number of pages $n_{pages \in C}$ included in $C$ and is described by the following formula:

$$Diversity(C) = \frac{n_{domains \in C}}{n_{pages \in C}} \quad (3)$$

Using the above metric we compared the datasets that were used in previous work (see Table 1). We can observe that the diversity of most of the datasets is quite low. In our experiments we demonstrate the performance of the SD Algorithm in a diverse dataset (SD Diverse) and in a less diverse dataset (SD Non-Diverse) similar to previous work and we show that SD Algorithm achieves very promising results in both of them. In addition, we demonstrate the importance of addressing the diversity in the evaluation since very good results for non-diverse datasets may be misleading.

| Dataset | Domains | Pages | Diversity (%) |
|---|---|---|---|
| Yi 2003 [19] | 5 | 5469 | 0.091 |
| Vieira 2006 [16] | 5 | 5259 | 0.095 |
| Cao 2008 [4] | 100 | 25910 | 0.385 |
| Nam 2009 [15] | 10 | n/a | n/a |
| Zhang 2010 [21] | 3 | 1500 | 0.200 |
| SD Non-Diverse | 10 | 5400 | 0.185 |
| SD Diverse | 521 | 600 | **86.833** |

**Table 1: Comparison of the diversity of the datasets from previous work compared to the datasets that we used for the evaluation of the SD Algorithm.**

## 4.3 Evaluation metrics

The evaluation for the classification task is based on the well-known measures of precision and recall. The recall metric was not very meaningful for some of our experiments due to the overlap between some of the classes, e.g. for the 3-class classification task. Furthermore, in order to evaluate the detection of the regions, we developed a "region accuracy metric" which is calculated by the following formula:

$$RA_{metric} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{k_i}\sum_{j=1}^{k_i}\frac{c_{ij}}{|c_{ij} - r_{ij}| + c_{ij}}\right) \quad (4)$$

where $n$ is the number of documents, $k_i$ is the number of regions with informative text inside document $i$, $c_{ij}$ is the density of the detected region $j$ in document $i$ (in case that region $j$ is not detected then $c_{ij}$ is zero) and $r_{ij}$ is the real (or actual) density of region $j$ in document $i$ (based on the annotation). The density is calculated by the number of characters in a region (see Section 3.1). Intuitively, this metric penalizes the algorithm if the extracted region is larger or shorter in comparison to the human annotated region. The score ranges between 0 to 1 and the best score is achieved for exact region capture with value 1.

# 5. EXPERIMENTAL RESULTS

In this section we report the experiments conducted to acquire the optimal parameter settings of our algorithm and evaluate its effectiveness on the extraction of informative textual parts from web pages. Furthermore, we study the ability of the method to distinguish between the three main web page types used in this study using both SD Diverse and SD Non-Diverse datasets.

## 5.1 Learning rate

In this first experiment we examine the learning abilities of SD Algorithm with respect to its statistical thresholds (T1 values range in [0-100] and T2 values are greater or equal to zero). The goal of this experiment is to find which is the appropriate size of the dataset to be used as training set for the learning of the parameters. In Fig. 3 we conducted a set of experiments increasing the amount of data from 5 to 200 for each of the three classes using extreme values for the two thresholds: $T2 = 20$ for all classes, $T1 = 100$ for Multiple areas and $T1 = 0$ for Articles and Articles with Comments. On the y-axis the average precision of 10 random selection runs is displayed and on the x-axis the number of data used. We observe that after 40 pages the average precision is stabilized and no further improvement is made. Therefore, we could use this portion of the dataset (20%) in order to calculate the optimal parameters of the algorithm.

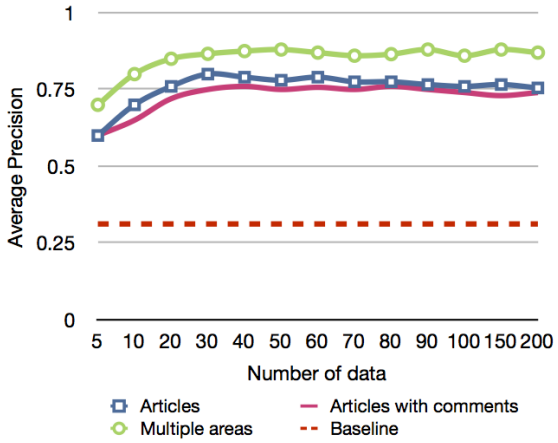


**Figure 3: The learning rate of the algorithm.**

| | SD Diverse dataset | | | SD Non-Diverse dataset | | |
|---|---|---|---|---|---|---|
| **Classes \ Evaluation metrics** | Precision | Recall | $RA_{metric}$ | Precision | Recall | $RA_{metric}$ |
| Articles (A) | 0.75 | 1 | 0.82 | 0.84 | 1 | 0.91 |
| Articles with Comments (AC) | 0.74 | 1 | 0.82 | 0.85 | 1 | 0.92 |
| Multiple areas (M) | 0.87 | 1 | 0.75 | 0.98 | 1 | 0.95 |
| A vs. AC vs. M | 0.74 | - | 0.78 | 0.81 | - | 0.89 |

**Table 2: Classification and informative region extraction results.**

## 5.2 Optimal Statistical Thresholds

Using a random 20% of the diverse dataset as explained on the learning rate subsection, we conducted a set of experiments in order to demonstrate the optimality of statistical thresholds for each of the classes. In Fig. 4, 5 the performance (average precision of 10 runs) of the SD Algorithm (y-axis) is depicted for a range of T1 and T2 values respectively (x-axis) for each of the classes examined. Recall that the T1 threshold is the max allowed difference from max density region and is used to detect candidate article regions while the T2 threshold is the minimum allowed density for a region in order not to be treated initially as noise. A baseline algorithm is based on random guess of class assignment.

In Fig. 4 we can observe the variation of T1 values while keeping T2 threshold fixed (T2=20). The low values (0-40) of threshold T1 favor the first two classes (Articles and Articles with Comments) because there is only one candidate article region. In contrast, the high values (60-100) of threshold T1 favor the third class (Multiple areas) due to the higher number of candidate article regions. Using the middle values of T1 (40-60), we can observe that all three classes have similar average precision.

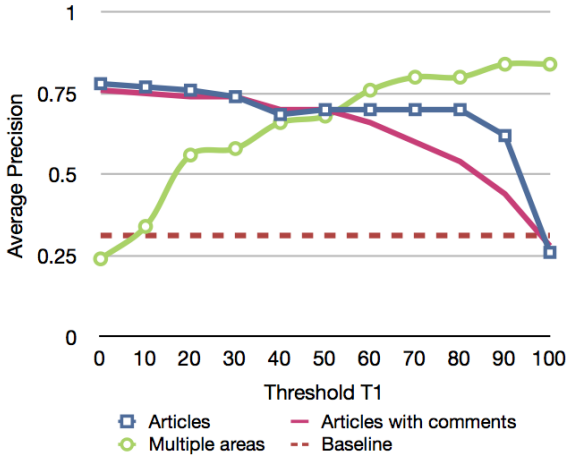


**Figure 4: Optimal threshold T1.**

In Fig. 5 the variation of T2 values are displayed while keeping fixed the optimal values of T1 for each class that were calculated previously. We observe that the T2 threshold has more predictable effect on the performance, the values ranging from 0 to 100 are the optimal for all the three classes. As the value of T2 is increased then initially many regions are considered as noise, thus the performance of the algorithm decreases dramatically after the value of 200. The multiple areas have a high value there due to the fact that empty multiple regions are returned as a result. We are dealing with three classes so the algorithm makes a decision based on the given regions extracted. In order to avoid such cases a low value of T2 as optimal (T2=20) is more appropriate, since it is more meaningful; noisy regions usually consist of low density regions.

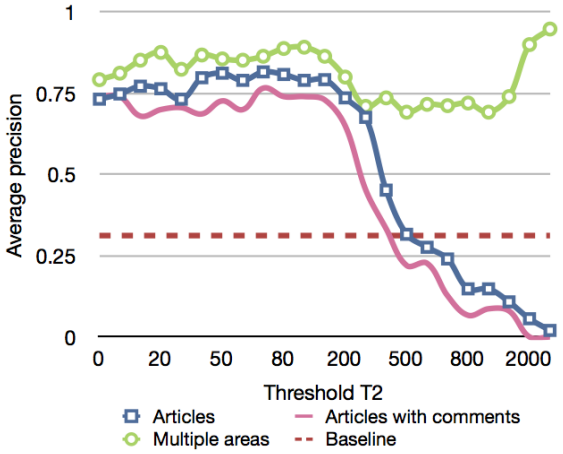


**Figure 5: Optimal threshold T2.**

## 5.3 SD Algorithm as a Rule-based Classifier

Based on the above experiments we performed 1-class and 3-class classification tasks for both SD Diverse and SD Non-Diverse datasets using 5-fold cross validation (20% training, 80% testing). The results of the classification and the informative region extraction are summarized in Table 2. We can observe that in both datasets the accuracy in extracting informative regions ($RA_{metric}$) is greater than the precision on the classification task. This happens due to the overlapping features between the classes. Even if an Article is misclassified as an Article with Comments and vice versa, the extraction of the article region is still correct. This overlap between these classes can be derived from the comparison of the precision and the $RA_{metric}$ values in both datasets.

The extraction of the informative regions for the Multiple areas class is less effective in the diverse dataset than in the non-diverse. The diversity in the dataset introduces difficulties in detecting the exact annotated informative regions due to the variety of different templates. Nevertheless, the overall results on extracting the informative regions when the class is not known initially (3-class setting) is promising in both datasets (78% and 89% respectively). Moreover, the classification performance of the examined classes is relatively high (74% and 81%) considering the difficulties of the 3-class classification task. Finally, the experimental results on two different datasets demonstrate the ability of the SD Algorithm to perform well in a realistic web setting.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper we presented a new approach to extract informative textual parts from web pages. Our method applies two basic pre-processing steps, namely web page segmentation and noise removal that are necessary in web driven tasks and applications. The proposed SD Algorithm combines visual and non-visual characteristics of web pages and is able to identify the type of web pages according to the properties of the detected regions. A series of experiments based on two annotated corpora of web pages has demonstrated the effectiveness of the SD Algorithm in extracting informative textual parts from three major types of web pages in a realistic web setting. Moreover, we demonstrated that the learning of the optimal parameter values can be calculated in a small subset of the given dataset. We also have shown how the SD Algorithm can be used as a rule-based classifier to distinguish between these web page types and the importance of the diversity of the datasets during evaluation. Finally, we plan to further evaluate this work in the framework of an opinion retrieval and mining system. Another future work direction is the exploitation of machine learning algorithms to handle the merging of areas and the characterization of different types of web pages.

## 7. ACKNOWLEDGEMENTS

The work described in this article was supported by the European Union through the inEvent project FP7-ICT n. 287872 (see http://www.inevent-project.eu). The authors thank Andrei Popescu-Belis and Thomas Meyer for their helpful remarks.

## 8. REFERENCES

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

[2] S. Baluja. Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. *Proceedings of the 15th international conference on World Wide Web*, pages 33–42, Edinburgh, Scotland, UK, 2006.

[3] D. Cai, S. Yu, and J.-r. Wen. VIPS : a Vision-based Page Segmentation Algorithm. *Microsoft Technical Report (MSR-TR-2003-79)*, 2003.

[4] D. Cao, X. Liao, H. Xu, and S. Bai. Blog post and comment extraction using information quantity of web format. In *Proceedings of the 4th Asia information retrieval conference*, AIRS'08, pages 298–309, Berlin, Heidelberg, 2008.

[5] D. Chakrabarti, R. Kumar, and K. Punera. Page-level template detection via isotonic smoothing. *Proceedings of the 16th international conference on World Wide Web*, pages 61–70, Banff, Alberta, Canada, 2007.

[6] Y. Chen, W.-Y. Ma, and H.-J. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. *Proceedings of the 12th international conference on World Wide Web*, pages 225–233, New York, USA, 2003.

[7] I. Debnath, P. Mitra, N. Pal, and C. L. Giles. Automatic identification of informative sections of web pages. *TKDE*, pages 1233–1246, 2005.

[8] Y. Diao, H. Lu, S. Chen, and Z. Tian. Toward learning based web query processing. *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 317–328, San Francisco, CA, USA, 2000.

[9] D. Gibson, K. Punera, and A. Tomkins. The volume and evolution of web page templates. *Special interest tracks of the 14th international conference on World Wide Web*, pages 830–839, New York, NY, USA, 2005.

[10] A. Java, P. Kolari, T. Finin, A. Joshi, J. Martineau, and J. Mayfield. Blogvox: Separating blog wheat from blog chaff. In *In Proceedings of the Workshop on Analytics for Noisy Unstructured Text Data, 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, Hyberabad, India, 2007.

[11] J. Kang, J. Yang, and J. Choi. Repetition-based web page segmentation by detecting tag patterns for small-screen devices. *IEEE Transactions on Consumer Electronics*, 56(2):980–986, 2010.

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

[13] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Rec.*, 31:84–93, 2002.

[14] S.-H. Lin and J.-M. Ho. Discovering informative content blocks from web documents. *Proceedings of the 8th international conference on Knowledge discovery and data mining (SIGKDD)*, 2002.

[15] S.-H. Nam, S.-H. Na, Y. Lee, and J.-H. Lee. Diffpost: Filtering non-relevant content based on content difference between two consecutive blog posts. In *Advances in Information Retrieval*, pages 791–795. Springer, Berlin, Heidelberg, 2009.

[16] K. Vieira, A. S. da Silva, N. Pinto, E. S. de Moura, J. a. M. B. Cavalcanti, and J. Freire. A fast and robust method for web page template detection and removal. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, New York, NY, USA, 2006.

[17] G. Vineel. Web page dom node characterization and its application to page segmentation. In *Proceedings of the 3rd IEEE international conference on Internet multimedia services architecture and applications*, IMSAA'09, NJ, USA, 2009.

[18] H. Yan and M. Miao. Research and implementation on multi-cues based page segmentation algorithm. *International Conference on Computational Intelligence and Software Engineering, 2009. CiSE 2009.*, pages 1–4, 2009.

[19] L. Yi, B. Liu, and X. Li. Eliminating noisy information in Web pages for data mining. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, page 296, 2003.

[20] A. Zhang, J. Jing, L. Kang, and L. Zhang. Precise web page segmentation based on semantic block headers detection. pages 63–68, 2010.

[21] Y. Zhang and K. Deng. Algorithm of web page purification based on improved dom and statistical learning. *2010 International Conference on Computer Design and Applications (ICCDA)*, 2010.