DOM-Based Online Store Comments Extraction

Julián Alarte[a], Carlos Galindo[b], Carlos Martín[c] and Josep Silva[d]
*Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València,
Camí de Vera s/n, 46022 València, Spain
{jualal, cargaji, cmarabe1, josilga}@upv.edu.es*

Keywords: Information Retrieval, Web Mining, Block Detection, Comments Extraction, Consumer Reviews Extraction.

Abstract: Online stores often include a customer comments section on their product pages. This section is valuable for other customers, as they can read reviews from users who have previously purchased or tried the products. This feedback is also important for the owners and managers of online stores, as they can obtain valuable information about the products they sell, such as buyer opinions and ratings. Additionally, the comments section holds significant value for the manufacturers of the products, as they can analyze comments posted on various online stores to receive valuable feedback about their products. This work presents a novel technique to automatically extract from a web page the customer comments without knowing a priori the web page structure. The technique not only extracts text but also other types of relevant content, such as images, animations, and videos. It is based on the DOM tree and only needs to load a single web page to extract its product comments; therefore, it can be used in real-time during browsing without the need for page preprocessing. To train and evaluate the technique, we have built a benchmark suite from real and heterogeneous web pages. The empirical evaluation shows that the technique achieves an average F1 score of 90.4% and reaches 100% on most web pages.

1 INTRODUCTION

The exponential growth of the Internet and, consequently, the vast array of product options available, increases the cost of evaluating these options for the customer (Ursu, 2018). As a result, product reviews or comments provided by other customers can facilitate this process. Heinonen defines online product reviews as a way for business managers and customers to connect with other customers (Heinonen, 2011). It is a fact that customer comments on a website or application can influence another customer's purchasing decision (Johan, 2021). Indeed, various factors can influence a purchasing decision, such as the website quality (Aren et al., 2013), or the product rating (Hossin et al., 2019).

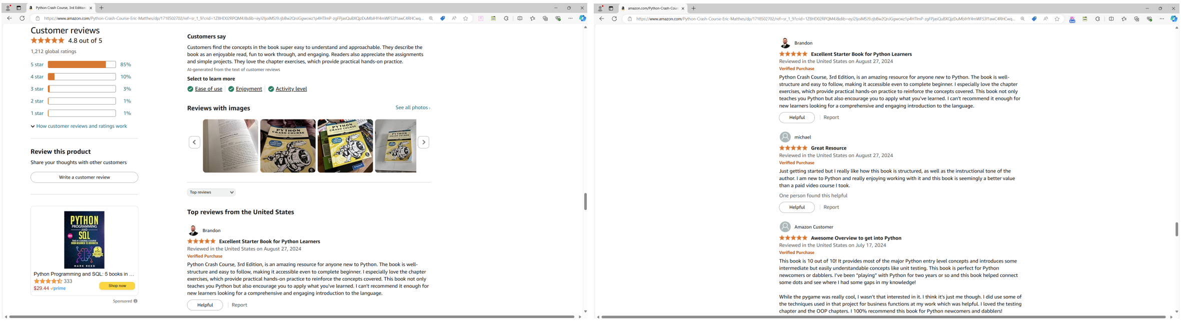
Sentiment analysis and opinion mining algorithms analyze consumer reviews or comments online to infer valuable information. Such information can be quite diverse. For example, Binali et al. (Binali et al., 2009) propose a classification into six groups: element extraction, feature extraction, element sentiment, feature sentiment, element comparison, and feature comparison. However, most of these techniques do not include a customer comment extraction process but rather treat them as resources to analyze, ignoring the problem of extracting them from the web. Some techniques extract the comments using scraping techniques (see, for example, (Chen et al., 2012; Saumya et al., 2018)), while others, such as the one proposed by Hu and Liu (Hu and Liu, 2004) and some extensions of it (Ding et al., 2008; Liu et al., 2015), use annotated datasets. From an engineering point of view, a web page is a set of nodes in the Document Object Model (DOM) (Consortium, 1997). Consequently, user comments or reviews on a web page can be defined as a subset of those nodes. It should be noted that user comments (and, therefore, those DOM nodes) typically contain not only text but also multimedia information such as images or videos (see Figure 1).

Our approach to comment extraction is based on analyzing the structures of the DOM model to represent web pages. Specifically, given a web page of an arbitrary product from an online store that includes comments, (1) we assign values to various features of some of its DOM nodes. (2) We compare the values computed for the features and the DOM nodes with equal values are grouped. (3) We remove groups that

[a] https://orcid.org/0009-0006-9825-3066
[b] https://orcid.org/0000-0002-3569-6218
[c] https://orcid.org/0009-0006-3973-1312
[d] https://orcid.org/0000-0001-5096-0008

Alarte, J., Galindo, C., Martín, C. and Silva, J.
DOM-Based Online Store Comments Extraction.
DOI: 10.5220/0012894600003825

In *Proceedings of the 20th International Conference on Web Information Systems and Technologies (WEBIST 2024)*, pages 15-24
ISBN: 978-989-758-718-4; ISSN: 2184-3252


Python Crash Course, 3rd Edition
https://www.amazon.com/Python-Crash-Course-Eric-Matthes/dp/1718502702/ref=sr_1_9?crid=1Z8HD02RPQM4J&dib=eyJ2IjoiMSJ9.rjbBw2QnJGgwcwz1p4HTlmP-zgFPjasQuBXQjzDuMbIHY4mWFS3f1awC4RHCwq...
Customer reviews
4.8 out of 5
1,212 global ratings
5 star 85%
4 star 10%
3 star 3%
2 star 1%
1 star 1%
How customer reviews and ratings work
Review this product
Share your thoughts with other customers
Write a customer review
Python Programming and SQL: 5 books in ...
333
$29.44 prime
Shop now
Sponsored
Customers say
Select to learn more
Ease of use
Enjoyment
Activity level
Reviews with images
See all photos ›
Top reviews
Top reviews from the United States
Helpful
Report

Customers find the concepts in the book super easy to understand and approachable. They describe the book as an enjoyable read, fun to work through, and engaging. Readers also appreciate the assignments and simple projects. They love the chapter exercises, which provide practical hands-on practice.

AI-generated from the text of customer reviews

Brandon

Excellent Starter Book for Python Learners

Reviewed in the United States on August 27, 2024

Verified Purchase

Python Crash Course, 3rd Edition, is an amazing resource for anyone new to Python. The book is well-structure and easy to follow, making it accessible even to complete beginner. I especially love the chapter exercises, which provide practical hands-on practice to reinforce the concepts covered. This book not only teaches you Python but also encourage you to apply what you've learned. I can't recommend it enough for new learners looking for a comprehensive and engaging introduction to the language.

michael

Great Resource

Reviewed in the United States on August 27, 2024

Verified Purchase

Just getting started but I really like how this book is structured, as well as the instructional tone of the author. I am new to Python and really enjoying working with it and this book is seemingly a better value than a paid video course I took.

One person found this helpful

Amazon Customer

Awesome Overview to get into Python

Reviewed in the United States on July 17, 2024

Verified Purchase

This book is 10 out of 10! It provides most of the major Python entry level concepts and introduces some intermediate but easily understandable concepts like unit testing. This book is perfect for Python newcomers or dabblers. I've been "playing" with Python for two years or so and this book helped connect some dots and see where I had some gaps in my knowledge!

While the pygame was really cool, I wasn't that interested in it. I think it's just me though. I did use some of the techniques used in that project for business functions at my work which was helpful. I loved the testing chapter and the OOP chapters. I 100% recommend this book for Python newcomers and dabblers!

Figure 1: Product page from amazon.com (left) and its comments extracted with our technique (right).

contain fewer nodes. (4) For each remaining group, we compute the closest common ancestor of all its members. (5) We analyze the obtained ancestors to identify which one corresponds to the root of the web page's comments section.

The main contributions of this work are: (i) defining the features shared by all comments on a web page and the comment extraction algorithm based on them; (ii) implementing the technique as an open-source WebExtension, which is an add-on officially evaluated and published by Mozilla Firefox on its web portal; and (iii) a benchmark suite containing heterogeneous web pages with user comments extracted from real websites. These pages have been labeled for automatic processing, indicating in their own HTML code which parts of the page are user comments and which are not.

2 STATE OF THE ART

Web page mining is a discipline that deals with isolating different functional blocks of a web page. Therefore, this discipline includes techniques such as content detection, template extraction, menu detection, comment extraction, etc. There are different approaches to web page mining (see, for example, (Alarte and Silva, 2021; Faty et al., 2020; Kumar et al., 2022; Zhang et al., 2021)), and even there was a competition called *CleanEval* (Baroni et al., 2008), which included a dataset and a gold standard to evaluate template detection techniques.

A *pagelet* is "a self-contained logical region within a page that has a well-defined theme or functionality" (Bar-Yossef and Rajagopalan, 2002). Block detection techniques try to identify pagelets, and can be classified depending on how they internally represent web pages:

i. Approaches based on HTML code (Jamshed et al., 2019; Leonhardt et al., 2020; Xie et al., 2020) use the text of the web page. Many of them assume that the main content on a web page has a high density of text and a low density of HTML tags.

ii. Another approach involves using a rendered image of the web page in the browser. These techniques (for example (van den Akker et al., 2019)) are based on the assumption that the main content of a web page is usually located in its middle section, and all or part of it is visible to the user without scrolling. The main disadvantage of this type of technique is the need for rendering web pages, which is a computationally expensive operation.

iii. Currently, the most widespread approach is to use the representation of a web page as a DOM tree (Alarte and Silva, 2022b; Kumar et al., 2022; Shah et al., 2019). Our technique uses this representation to infer the information needed to compute the subtree corresponding to the comments section of the web page, i.e., the pagelet that corresponds to the comments section.

Typically, sentiment analysis and user opinion mining algorithms take sets of reviews or opinions obtained through scraping techniques specifically designed for each website as input. In contrast, our algorithm performs the automatic extraction of user comments from heterogeneous websites without prior knowledge of the web page structure and without the need for any pre-processing. This is possible because the algorithm is independent of the web page containing the reviews to be extracted. Therefore, our technique can be especially useful when combined with sentiment analysis and opinion mining techniques, as it can extract customer comments or reviews from different web pages regardless of their structure or any changes in their content.

Moreover, this technique is not only useful in combination with sentiment analysis techniques but also in combination with other types of web page mining techniques. Some of these techniques require the identification of a specific block of a web page to work properly. For example, opinion mining techniques (Hu and Liu, 2004) require identifying the opinion block, i.e., the block of a web page that contains user opinions. Our approach could be used as a previous step to automatically identify this block on any web page.

3 COMMENTS SECTION ON A WEB PAGE

For a human, identifying the comments section in a rendered view of a product web page is trivial. The comments section is usually within the main content of the web page and typically contains several paragraphs of text related to the product in a repetitive structure. However, when a web page is represented as a DOM tree instead of its rendered view, identifying the comments section becomes a complex task. Due to the hierarchy of the DOM tree, a particular fragment of a web page is often contained in several of its DOM nodes, making it challenging to determine which one adequately represents the set of comments. Additionally, a web page usually contains various text blocks in different fragments (e.g., the comments section, a description, an ad, etc.) that can be arbitrarily separated in the DOM tree.

The formal definition of the comments section in a DOM tree is based on the formal definition of a web page. For convenience, this definition only specifies the types of nodes that are relevant to the proposed technique.

Definition 1 (Web Page (Alarte and Silva, 2021))**.** *A* web page *P is a tree (N,A) formed from a finite set of nodes N. Each non-leaf node $n \in N$ is of type element ($n.nodeType = 1$)*[1] *and contains an HTML tag (including its attributes). Leaf nodes can be text nodes ($n.nodeType = 3$), CDATA section nodes ($n.nodeType = 4$), or comment nodes ($n.nodeType = 8$).*[2] *The root node corresponds to the HTML* `body` *tag. A is a finite set of edges such that $(n \to n') \in A$, with $n, n' \in N$, if and only if the tag or text associated with n' is inside the tag associated with n, and there is no unclosed tag between them.*

Given a node n on a web page P, $\textsc{descendants}(n)$ consists of all nodes belonging to the subtree of n excluding n, and $\textsc{subtree}(n)$ consists of the union of n and the nodes forming $\textsc{descendants}(n)$. The set of leaf nodes in $\textsc{descendants}(n)$ is obtained by $\textsc{leaves}(n)$, while $\textsc{childNodes}(n)$ represents the set of children of n. The parent node of a node n can be accessed through $\textsc{parent}(n)$. $\textsc{depth}(n)$ is the depth of node n (measured as the number of edges from the `body` node to the node), and $\textsc{maxDepth}(P)$ is the maximum depth of all nodes in P. The total number of text words (excluding those belonging to hyperlinks) in $\textsc{descendants}(n)$ is $\textsc{words}(n)$.

Next, a formal definition of *comments section* is introduced, which is exclusively based on the structure of the web page and, therefore, independent of any detection method. Given a web page $P = (N,A)$, we assume the existence of a $\textsc{comment}(n)$ tag for leaf nodes that identifies those nodes in the web page that correspond to user comments (both textual and non-textual information: multimedia, etc.).

Definition 2 (Comments Section)**.** *The* comments section *of a web page $P = (N,A)$ is a set of DOM nodes $C \subset N$ such that:*

i. All nodes that are user comments belong to the subtrees of the nodes in the comments section: $\forall\, n \in N :\ \textsc{comment}(n) \implies n \in \textsc{subtree}(n' \in C)$.

ii. All leaf nodes that belong to the subtrees of the nodes in the comments section are user comments: $\forall\, n \in \textsc{leaves}(n' \in C) : \textsc{comment}(n)$.

iii. The set of nodes in the comments section is minimal: $\nexists\ C' \subset C\ .\ \forall\ n \in N,\ \textsc{comment}(n),\ n \in \textsc{subtree}(n' \in C')$.

It should be noted that Definition 2 is useful when the labeling $\textsc{comment}(n)$ can be provided. If the labeling $\textsc{comment}(n)$ is not available, it should be approximated. For example, product comments extraction algorithms are useful tools for this, as they automatically create the $\textsc{comment}(n)$ labeling.

4 COMMENTS SECTION EXTRACTION

In this section, we propose a technique for extracting product comments based on the classification of DOM nodes throughout a set of features. The input data for the technique is a web page from an e-commerce site containing product comments, and the output is a DOM node corresponding to the root of the comments section. Since the technique is applied to a web page (not a website), it only needs to load and analyze a single web page (the input page) to infer the comments section. This is particularly important as it is directly related to the algorithm's runtime.

The technique is divided into five phases:

[1] In the DOM tree, all nodes are labeled with the *nodeType* attribute.

[2] Henceforth, it is crucial not to confuse DOM nodes *of comments*, which introduce developer comments on the web page, with DOM nodes *representing comments* from the user on the page.

i. Some nodes of the DOM tree are selected using an algorithm, and for each of them, four values are computed: number of leaves, number of descendants, number of children, and depth of the DOM node.

ii. Next, the nodes are classified into groups of nodes with exactly the same values.

iii. Those groups containing a number of nodes less than an empirically calculated threshold are removed.

iv. For each remaining group, an algorithm computes the nearest common ancestor of all its nodes.

v. Finally, for each ancestor calculated in the previous phase, an algorithm computes a value based on four properties (words, groups, depth, images—these properties are explained in Section 4.5) that the comments section should have. The ancestor with the highest value corresponds to the root DOM node of the comments section.

All proposed ideas have been empirically validated with a set of tests used to parameterize the algorithms (see Section 5). The following sections describe each phase in depth.

4.1 Property Assignment

This subsection introduces a metric to identify those DOM nodes that potentially correspond to the product comments section of an e-commerce site.

First, we explore the DOM tree of the web page to compute and assign a value to each node that meets the following criteria:

i. It is not a leaf node of the DOM tree.

ii. It is an element node and its *tagName* is different from the following: *ignoredTags* = {`a`, `body`, `button`, `em`, `form`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `header`, `hr`, `iframe`, `img`, `label`, `nav`, `noscript`, `option`, `script`, `select`, `style`, `undefined`}.

iii. $\text{DEPTH}(n) + d \leq \text{MAXDEPTH}(P)$, where n is the node, P is the web page, and $d = 2$ (the value of the constant d has been obtained empirically, see Section 5).

The comments section of a product web page generally contains a high text density, but normally, this density is not enough to detect it. Next, we propose several properties that should be considered to properly detect the comments section. All these properties are objectively quantified and appropriately combined to form a value that can be used to identify the comments section of a product web page.

Definition 3 (Properties of a DOM node)**.** *Let* $P = (N, A)$ *be a web page. Each node* $n \in N$ *has the following properties if* $\text{DESCENDANTS}(n) = \emptyset$*,* $n.tagName \notin ignoredTags$*, and* $\text{DEPTH}(n) + d \leq \text{MAXDEPTH}(P)$*.*

Text Nodes: *The number of text nodes among the descendants of n:* $|\{m \mid m \in \text{DESCENDANTS}(n) \wedge m.nodeType = 3\}|$.

Size: *The number of descendants of n plus one (itself):* $|\text{SUBTREE}(n)|$.

Children: *The number of direct children of n:* $|\text{CHILDNODES}(n)|$.

Depth: *The depth of n in the DOM tree:* $\text{DEPTH}(n)$.

The rationale behind these properties is based on the fact that the comments section is typically composed of a set of repetitive DOM nodes (comments) that share the same structure. Therefore, on a web page, the root node of one comment is similar to the root node of another comment, as they share several properties. We observed that the properties in Definition 3 are commonly shared by the root nodes of comments. Thus, DOM nodes that share the same values for these properties are more likely to belong to the same section of the web page. However, this section is not necessarily the comments section and, for this reason, a further treatment is needed.

Once computed, these properties are assigned to each node of the DOM tree that satisfies the criteria described above.

Definition 4 (Equivalence of DOM nodes)**.** *Two nodes* $n_1, n_2 \in N$ *with quadruples* $n_1.prop = (a, b, c, d)$*, and* $n_2.prop = (a', b', c', d')$ *are equivalent if and only if* $a = a' \wedge b = b' \wedge c = c' \wedge d = d'$*, where prop is a quadruple containing the values of the four properties assigned to that node in Definition 3.*

4.2 Node Clustering

In this phase, the DOM nodes are clustered based on Definition 4. Groups of nodes with equivalent properties are created.

Algorithm 1 iterates through all nodes and classifies them into groups, where each group consists of equivalent DOM nodes based on Definition 4. It should be noted that in a web page where this grouping is performed, the variance in group sizes is typically large.

4.3 Removal of Groups with Fewer Nodes

In this phase, groups of nodes in the DOM tree that are less numerous are discarded. The number of nodes

```
Data: A set of DOM nodes called
      candidateNodes.
Result: The set of nodes candidateNodes
        classified into groups and the total
        number of groups g.
forall $n \in candidateNodes$ do
    $n.group \leftarrow null$
end
$g \leftarrow 0$
forall $n_1 \in candidateNodes$ do
    if ($n_1.group = null$) then
        $n_1.group \leftarrow g$
        forall $n_2 \in candidateNodes$ do
            if ($n_1 \neq n_2 \wedge n_2.group = null$
              $\wedge n_1.prop = n_2.prop$) then
                $n_2.group \leftarrow n_1.group$
            end
        end
        $g \leftarrow g + 1$
    end
end
return $(candidateNodes, g)$
```

Algorithm 1: Node Clustering.

forming each group is computed, and then groups with a number of nodes less than or equal to a certain threshold value are removed. It is important to note that the number of groups remains unchanged; but some of them have lost all their members. This makes the process more efficient.

Algorithm 2 takes a parameter t as input, representing the size threshold. In our implementation, the value of t has been determined empirically. The process is detailed in Section 5.

```
Data: A set of DOM nodes candidateNodes,
      the total number of groups g, and the
      threshold value t.
Result: The set of nodes candidateNodes
        with the group property updated.
forall $i \in [0, g]$ do
    if ($t \geq$
      $|\{n \mid n \in candidateNodes \wedge n.group = i\}|$)
      then
        forall $n \in candidateNodes$ do
            if ($n.group = i$) then
                $n.group \leftarrow null$
            end
        end
    end
end
return candidateNodes
```

Algorithm 2: Removal of Groups with Fewer Nodes.

4.4 Minimum Common Ancestor Calculation

Once the less numerous node groups are eliminated, for each remaining group, the nearest common ancestor shared by all its nodes is computed. In other words, the ancestor containing all nodes which is located at the deepest possible depth in the DOM tree.

For each group, Algorithm 3 computes the deepest node in the DOM tree containing all members of the group. To achieve this, it selects the first group member and recursively explores its ancestors. Each time it explores an ancestor, it checks whether it contains all members of the group; if so, that ancestor is the minimum common ancestor. All functions used by the algorithm are defined in Section 3.

The output of the algorithm is a set of nodes, each being an ancestor of one of the groups. These nodes are candidates to be the root node of the comments section of the web page.

```
Data: A set of DOM nodes candidateNodes
      and the number of groups g.
Result: A set of nodes candidate to be the
        root of the comments section.
$candidateRoots \leftarrow \{\}$
forall $i \in [0, g]$ do
    $groupNodes \leftarrow$
      $\{n \mid n \in candidateNodes \wedge n.group = i\}$
    if ($|groupNodes| > 0$) then
        $node \leftarrow n \in groupNodes$
        $p \leftarrow$ PARENT$(node)$
        while
          ($groupNodes \nsubseteq$ DESCENDANTS$(p)$)
          do
            $p \leftarrow$ PARENT$(p)$
        end
        $candidateRoots \leftarrow candidateRoots \cup p$
    end
end
return candidateRoots
```

Algorithm 3: Minimum Common Ancestor Calculation.

4.5 Selection of the Comments Section Root Node

For each group's ancestor obtained in the previous phase, the values of various properties are computed. Subsequently, a metric is computed with the obtained values for these properties. Finally, the node with the highest value for the computed metric corresponds to the root node of the comments section.

Definition 5 details the properties applied to each of the ancestors to select the root of the comments

Table 1: Average F1 on the training set as a function of parameters t and d.

	DOM nodes						Words					
d\t	0	1	2	3	4	5	0	1	2	3	4	5
0	23.81 %	23.81 %	32.24 %	40.18 %	53.18 %	53.18 %	25.42 %	25.42 %	32.38 %	39.87 %	53.73 %	53.73 %
1	23.81 %	23.81 %	45.26 %	70.20 %	70.20 %	45.68 %	25.42 %	25.42 %	46.26 %	71.25 %	71.25 %	47.19 %
2	31.15 %	31.15 %	72.14 %	**87.14** %	67.18 %	47.06 %	33.82 %	33.82 %	72.38 %	**87.38** %	67.38 %	47.06 %
3	36.05 %	36.05 %	79.41 %	84.41 %	54.45 %	44.44 %	36.93 %	36.93 %	79.04 %	84.04 %	54.04 %	44.44 %
4	37.60 %	37.60 %	65.06 %	54.41 %	44.45 %	33.33 %	39.62 %	39.62 %	66.25 %	54.04 %	44.04 %	33.33 %

section.

Definition 5 (Properties of the root node)**.** *Given a non-empty set of ancestor nodes of each group, N, each node n ∈ N is classified according to the following properties:*

Words: *The total number of words in all text nodes descending from n (except those contained in links):* $\text{WORDS}(n) = \sum \text{WORDS}(n)$.

Groups: *The number of* distinct *groups that can be found among the descendants of n:* $\text{GROUPS}(n) = |\{m.group \mid m \in \text{DESCENDANTS}(n)\}|$.

Depth: *The depth of n in the DOM tree:* $\text{DEPTH}(n)$.

Images: *The number of images existing among the descendants of n:* $\text{IMAGES}(n) = |\{m \mid m \in \text{DESCENDANTS}(n) \wedge m.tagName \in \{\texttt{img}, \texttt{svg}\}\}|$.

As in Section 4.1, the properties need to be combined to correctly differentiate the node representing the root of the comments section from the others. Definition 6 combines the values obtained for the different properties to produce a metric indicating how likely this group is the comments section.

Definition 6 (Root Node Selection Metric)**.** *Given a node n, the root of a group, and its properties calculated using Definition 5, its weight is:*

$$\text{WEIGHT}(n) = \frac{\text{WORDS}(n) \times \text{GROUPS}(n) \times \text{DEPTH}(n)}{\text{IMAGES}(n)}$$

Finally, we define a heuristic to select the root node representing the comments section: the node from the set of ancestor nodes of each group with the highest weight (according to Definition 6) corresponds to the root node of the comments section.

5 EMPIRICAL EVALUATION

This technique has been implemented as a WebExtension[3], compatible with Firefox or Chromium-based browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera, among others. The technique's implementation has been evaluated by Mozilla and published on its official extension portal for Firefox[4]. Before publication, Mozilla requires several rounds of review to ensure the quality and security of published extensions. Users of other browsers can download the extension from its website[5]. This WebExtension appears as a single button in the browser. When pressed, it extracts the comments section of the current web page, which is displayed automatically[6] (and can be saved). If the button is pressed again, the original web page is shown.

We conducted a battery of experiments using real and heterogeneous online web pages to measure the performance of the technique. For this, we measured *recall*, *precision*, and *F1* (Gottron, 2007). Recall is computed as the number of nodes (or words) correctly obtained divided by the number of nodes (or words) in the comments section. Precision corresponds to the number of nodes (or words) correctly obtained divided by the total number of nodes (or words) obtained. F1 is computed as $2PR/(P+R)$, where R is recall and P is precision.

Initially, we tried to use a standard public collection of test web pages, but we found no suitable public dataset for customer comments extraction. Some of them were not prepared for DOM-based techniques or included very few products (Hu and Liu, 2004), while others only included comments from a single web page (Ni et al., 2019). We also could not adapt a set of test web pages prepared for template and content extraction, such as TeCo (Alarte and Silva, 2022a), as it included very few product web pages with user comments. Therefore, we decided to create one of the most important contributions of our work: a new test set prepared for user comments extraction that is publicly available at https://mist.dsic.upv.es/revEx/downloads.html.

We created a dataset with 50 real and heterogeneous product web pages from different websites. As

[3] https://developer.mozilla.org/en/docs/Mozilla/Add-ons/WebExtensions

[4] https://addons.mozilla.org/en/firefox/addon/review-extractor/

[5] https://mist.dsic.upv.es/revEx/downloads.html

[6] DOM nodes that do not belong to the comments section are properly hidden by changing their *visibility* and *display* attributes to *hidden* or *none*, respectively. Therefore, the comments section is isolated and appears in the same place as on the original web page.

Table 2: Empirical Evaluation of the Technique.

	DOM nodes			Words			
Domain	**Recall**	**Precision**	**F1**	**Recall**	**Precision**	**F1**	**Runtime**
aftfasteners.com	100.00 %	15.42 %	26.72 %	100.00 %	11.91 %	21.28 %	0.27 s.
amazon.es	96.47 %	100.00 %	98.20 %	98.60 %	100.00 %	99.30 %	1.88 s.
beechworthhoney.com.au	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.10 s.
bergfreunde.de	89.91 %	100.00 %	94.69 %	90.56 %	100.00 %	95.05 %	1.23 s.
bluschoolsupplies.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.12 s.
decathlon.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.65 s.
dosfarma.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	1.48 s.
druni.es	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.12 s.
eltallerdelmodelista.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.04 s.
etsy.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.71 s.
farmaciajimenez.com	91.84 %	100.00 %	95.75 %	94.61 %	100.00 %	97.23 %	0.15 s.
huntoffice.ie	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.36 s.
juegoyjardin.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.18 s.
majestic.co.uk	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.60 s.
matalan.co.uk	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.14 s.
musicstore.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.35 s.
mylittlewardrobe.com.au	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.87 s.
neobyte.es	97.42 %	100.00 %	98.69 %	99.47 %	100.00 %	99.73 %	0.34 s.
otto.nl	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.39 s.
parisfashionshops.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.34 s.
pharmabuy.es	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.09 s.
shopcoffee.co.uk	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	1.48 s.
snooplay.in	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.14 s.
sundae-muse.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.55 s.
tcompanyshop.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.12 s.
tea-and-coffee.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.16 s.
tshirtstudio.es	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.03 s.
watchard.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	1.40 s.
winechateau.com	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.49 s.
zonadepadel.es	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	0.31 s.
Average	92.52 %	90.51 %	90.47 %	92.77 %	90.40 %	90.42 %	0.50 s.

a result, all pages have different designs and structures, including web pages in different languages to validate the language independence of the technique. Each web page in the dataset consists of a single page, archived along with all its resources (e.g., CSS, JavaScript, videos, or images). This ensures the independence of each page regarding its evolution over time (design changes, data updates, etc.). For each web page in the dataset, the user comments section was manually obtained and then the *mainComments* class was added to the HTML tag of its root DOM node. This way, the dataset is useful for researchers to evaluate or compare their comment extraction techniques using that tag.

5.1 Training Phase

We built a version of our WebExtension to automate the evaluation. For all web pages in the dataset, it sequentially runs the comment extraction algorithm. For each test, it computes the recall, precision, F1, and execution time for both text words and obtained DOM nodes. The technique's training consists of determining the optimal value of two parameters to maximize F1:

- Value of the parameter d, which defines the maximum depth a DOM node should have concerning the maximum depth of the DOM tree to assign it the values of the properties in Section 4.1.
- Threshold value t (see Section 4.3) that defines the minimum number of elements a group must have to avoid being eliminated.

During the training phase, 20 web pages were randomly selected from the dataset to form a training subset, and recall, precision, and F1 were computed for different values of the two parameters. Table 1 shows the results of the experiments carried out with the training subset. The value of each cell in the table is the average F1 of the 20 experiments for each combination of the parameters. Possible values of d ranged from 0 to 4, while possible values of t ranged from 0 to 5. Each row in the table corresponds to a possible value of d, and each column corresponds to a possible value of t. It should be highlighted that the table shows the results obtained for both DOM nodes (first 5 columns of results) and text words (last 5 columns).

We can see that the best average F1 is obtained for the combination of $d = 2$ and $t = 3$, both for DOM nodes and text words.

5.2 Evaluation Phase

The evaluation dataset consisted of the remaining 30 web pages. Table 2 presents the results obtained for optimal parameters t and d. Each row in the table corresponds to a web page (column `Domain` refers to the domain of the web page). As shown in the table columns, the `Recall`, `Precision`, and `F1` were computed for both DOM nodes and extracted text words. Additionally, the `Time` in seconds was measured. The results show an average F1[7] of 90.47% for DOM nodes and an average F1 of 90.42% for text words. It should be noted that in 80% of the web pages, an F1 of 100% is achieved. However, there are two web pages where an F1 of 0% is obtained, which indicates no detection of any part of the comments section. We observed that this happens because the algorithm identifies a set of DOM nodes that share the properties in Definition 3, but these DOM nodes do not correspond to the comments of the web page. In the case of beachworthhoney.com.au, the algorithm identifies a set of DOM nodes that correspond to a set of other products from the webshop. On the other hand, in the case of mylittlewardrobe.com.au, the algorithm identifies as comments a set of DOM nodes that correspond to images of a carousel.

Finally, we observe that once the web page is loaded by the browser, the average algorithm time is 0.5 seconds, which is consistent with a page-level block extraction technique, as these techniques only need to perform operations on the web page from which they want to obtain the block.

5.3 Comparison with Other Techniques

We could not find in the literature any other comments extraction technique able to work with heterogeneous web pages. We found, however, some commercial tools that, among other things, allow us to extract reviews from product web pages. Unfortunately, most of these tools are not valid for heterogeneous web pages. Some of them are only prepared for particular web pages (e.g., Apify[8], or Outscraper[9]), while others take as input the HTML tags or blocks that contain the comments section (e.g., the iSocialWeb Product Review Extractor[10], or BrowseAI[11]). We also found two commercial tools that are able to extract comments from heterogeneous web pages, Zyte[12] and ScrapeStorm[13]. We compared both tools with *RevEx* using 30 product web pages from our dataset. *RevEx* successfully extracted the comments section from 27 web pages, while ScrapeStorm extracted it from 10 web pages and Zyte from 3 web pages. As a result, *RevEx* demonstrated a significantly higher efficacy compared to both commercial tools.

6 CONCLUSIONS

Opinion mining and sentiment analysis algorithms take sets of online consumer comments as input and analyze them to infer valuable information. These comments are often obtained through scraping techniques, which depend on the specific web page they are applied to. This work presents a new technique for extracting the comments section from heterogeneous product web pages.

The main novelty of our technique lies in defining the characteristics of DOM nodes that allow clustering them into groups of equivalent DOM nodes, and how the comments section is detected from these groups. The features considered in the metrics have proven to be useful for inferring the root node of the comments section on a product web page. Computing the features of DOM nodes and clustering them into groups of equivalent nodes is helpful in easily and quickly identifying nodes that follow a common structure.

Another significant contribution of the work is that we have created and made public a dataset consisting of 50 heterogeneous product web pages. This dataset allowed us to train and evaluate the technique.

The empirical evaluation shows that the technique achieves an average F1 score of over 90% for both, text words and DOM nodes. It should be highlighted that perfect results (exact extraction of the comments section with an F1 of 100%) are obtained on 80% of web pages. Regarding the average execution time of the algorithm, it is 0.5 seconds, excluding the web page loading time.

A strong point of our technique is that, unlike most block extraction techniques that only focus on extracting text, it can extract comments content regardless of its type. In other words, it not only extracts text

[7] Average F1 is the mean of the values in the F1 column, not F1 calculated using the mean precision and mean recall.

[8] https://apify.com

[9] https://outscraper.com

[10] https://www.isocialweb.agency/en/ai-ecommerce-product-review-extractor/

[11] https://www.browse.ai

[12] https://www.zyte.com

[13] https://www.scrapestorm.com

but also animations, images, and videos. Our implementation is open and free, available on the official Mozilla Firefox add-ons portal. The webExtension can be also combined with tools such as Selenium in order to automate the generation of product reviews from different websites.

As future work, we plan to incorporate other output formats for the webExtension (besides HTML). One of these formats will be plain text since, usually, sentiment analysis and opinion mining algorithms input the comments as plain text. On the other hand, the functionality of many scraping tools is based on knowing the *className* or the *id* of the DOM node. Therefore, another interesting output is the *className* and *id* of the DOM node that corresponds to the root of the comments section. We also plan to augment our dataset of product web pages with more real webpages labelled for comments extraction.

Finally, it should be highlighted that not all comments are always visible on a web page (sometimes the user has to press a button called “Show more” or similar), which is a limitation of our technique. However, we are investigating how to retrieve all comments, whether they are visible or not.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish MCIN/AEI under grant PID2019- 104735RB-C41 and by Generalitat Valenciana under grant CIPROM/2022/6 (Fasslow). Carlos Galindo was partially supported by the Spanish Ministerio de Universidades under grant FPU20/03861.

REFERENCES

Alarte, J. and Silva, J. (2021). Page-level main content extraction from heterogeneous webpages. *ACM Trans. Knowl. Discov. Data*, 15(6).

Alarte, J. and Silva, J. (2022a). A benchmark suite for template detection and content extraction.

Alarte, J. and Silva, J. (2022b). Hybex: A hybrid tool for template extraction. In *Companion Proceedings of the Web Conference 2022*, WWW ’22, page 205–209, New York, NY, USA. Association for Computing Machinery.

Aren, S., Güzel, M., Kabadayı, E., and Alpkan, L. (2013). Factors affecting repurchase intention to shop at the same website. *Procedia - Social and Behavioral Sciences*, 99:536–544. The Proceedings of 9th International Strategic Management Conference.

Bar-Yossef, Z. and Rajagopalan, S. (2002). Template detection via data mining and its applications. In *Proceedings of the 11th International Conference on World Wide Web (WWW’02)*, pages 580–591, New York, NY, USA. ACM.

Baroni, M., Chantree, F., Kilgarriff, A., and Sharoff, S. (2008). Cleaneval: a Competition for Cleaning Web Pages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC’08)*, pages 638–643.

Binali, H., Potdar, V., and Wu, C. (2009). A state of the art opinion mining and its application domains. In *2009 IEEE International Conference on Industrial Technology*, pages 1–6.

Chen, L., Qi, L., and Wang, F. (2012). Comparison of feature-level learning methods for mining online consumer reviews. *Expert Systems with Applications*, 39(10):9588–9601.

Consortium, W. (1997). Document Object Model (DOM). Available from URL: http://www.w3.org/DOM/.

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM ’08, page 231–240, New York, NY, USA. Association for Computing Machinery.

Faty, L., Ndiaye, M., Sarr, E. N., and Sall, O. (2020). Opinionscraper: A news comments extraction tool for opinion mining. In *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–5.

Gottron, T. (2007). Evaluating content extraction on HTML documents. In *Proceedings of the 2nd International Conference on Internet Technologies and Applications (ITA’07)*, pages 123–132. National Assembly for Wales.

Heinonen, K. (2011). Consumer activity in social media: Managerial approaches to consumers’ social media behavior. *Journal of Consumer Behaviour*, 10(6):356–364.

Hossin, M., Mu, Y., Fang, J., and Kofi Frimpong, A. N. (2019). Influence of picture presence in reviews on online seller product rating: Moderation role approach. *KSII TIIS*, 13.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Jamshed, H., Khan, S. A., Khurrum, M., Inayatullah, S., and Athar, S. (2019). Data preprocessing: A preliminary step for web data mining. *3c Tecnología: glosas de innovación aplicadas a la pyme*, 8(1):206–221.

Johan, A. (2021). Product ranking: Measuring product reviews on the purchase decision. *Business & Economic Review*, 4.

Kumar, A., Morabia, K., Wang, W., Chang, K., and Schwing, A. (2022). Cova: Context-aware visual attention for webpage information extraction. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*. Association for Computational Linguistics.

Leonhardt, J., Anand, A., and Khosla, M. (2020). Boilerplate removal using a neural sequence labeling model. In

Companion Proceedings of the Web Conference 2020, WWW '20, page 226–229, New York, NY, USA. Association for Computing Machinery.

Liu, Q., Gao, Z., Liu, B.-Q., and Zhang, Y. (2015). Automated rule selection for aspect extraction in opinion mining. In *International Joint Conference on Artificial Intelligence*.

Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Saumya, S., Singh, J. P., Baabdullah, A. M., Rana, N. P., and Dwivedi, Y. K. (2018). Ranking online consumer reviews. *Electronic Commerce Research and Applications*, 29:78–89.

Shah, H., Rezaei, M., and Fränti, P. (2019). Dom-based keyword extraction from web pages. In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, AIIPCC '19, New York, NY, USA. Association for Computing Machinery.

Ursu, R. M. (2018). The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science*, 37(4):530–552.

van den Akker, B., Markov, I., and de Rijke, M. (2019). Vitor: Learning to rank webpages based on visual features. In *The World Wide Web Conference*, WWW '19, page 3279–3285, New York, NY, USA. Association for Computing Machinery.

Xie, X., Fu, Y., Jin, H., Zhao, Y., and Cao, W. (2020). A novel text mining approach for scholar information extraction from web content in chinese. *Future Generation Computer Systems*, 111:859–872.

Zhang, M., Yang, Z., Ali, S., and Ding, W. (2021). Web page information extraction service based on graph convolutional neural network and multimodal data fusion. In *2021 IEEE International Conference on Web Services (ICWS)*, pages 681–687.