HybEx: A Hybrid Tool for Template Extraction

Julián Alarte
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
E-46022 Valencia, Spain
jalarte@doctor.upv.es

Josep Silva
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
E-46022 Valencia, Spain
jsilva@dsic.upv.es

ABSTRACT

HybEx is a site-level web template extractor that combines two algorithms for template and content extraction: (i) TemEx, a site-level template detection technique, and (ii) Page-level ConEx, a content extraction technique. The key idea is to add a preprocess to TemEx that removes the main content inferred by Page-level ConEx. It is a fact that adding this new phase to the TemEx algorithm involves an increase of its runtime, however, this increase is very small compared to the TemEx runtime because Page-level ConEx is a page-level technique. On the other hand, HybEx improves the precision, recall, and F1 of TemEx. This paper describes the new template extractor and its internal architecture. Furthermore, the paper also presents the results of its empirical evaluation.

CCS CONCEPTS

• **Information systems** → **Information extraction**; **Document filtering**; **Presentation of retrieval results**.

KEYWORDS

Information Retrieval, Template Extraction, Content Extraction, Web Mining, Block Detection

ACM Reference Format:
Julián Alarte and Josep Silva. 2022. HybEx: A Hybrid Tool for Template Extraction. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524242

1 INTRODUCTION

Template detection has multiple applications, and thus, it is possible to find several techniques for template detection and boilerplate removal (see, e.g., [1, 4, 9, 13, 15]). This paper presents HybEx, a tool that automatically extracts the template of a webpage by combining a template detection technique (TemEx [1]) with a content extraction technique (Page-level ConEx [2]). HybEx improves the results of TemEx, which was compared in [3] with 4 of the most advanced template extractors in the literature. The empirical evaluation results of the comparison showed that, currently, TemEx produces the best retrieval results for heterogeneous webpages (considering recall, precision, and F1). HybEx can work as a library for other systems such as crawlers, and it is also ready to be used by human users because it is implemented as a browser WebExtension (for Firefox, Chrome, and Chromium, among others).

The template elements in a webpage were measured by Gibson et al. [5], which estimated that they represent between 40% and 50% of all the data on the Web. Templates are important for the block detection discipline for many reasons. For instance, detecting templates is useful for boilerplate removal, component reuse, content detection, webpage generation, and webpage displaying for blind people, among many other applications.

There exist in the literature many template detection techniques (see e.g., [7, 14, 16]) and main content extraction techniques (see, e.g., [2, 8, 11]). However, despite many researchers have been working in the field for the last 15 years, we have not found any hybrid technique that combines template detection and content extraction algorithms. Not even the latest block detection techniques (see, e.g., [10, 14]) implement another block detection preprocess phase. Many techniques implement simple preprocess methods such as removing nodes that surely do not have any content to extract [14], or tag clipping and correction of syntax errors in HTML code to improve the results of subsequent phases [17]. However, despite not combining several block detection techniques, some authors based their techniques on the combination of several methods or different kinds of information. For instance, Song et al. [10] proposed a hybrid content extraction approach based on the combination of their measure of the text density (called textual information), and a visual measure for the evaluation of tags in webpages (called visual importance). Uzun et al. [12] proposed a hybrid method for extracting relevant content which is divided into two phases: the first one uses a machine learning method to discover informative content from the webpage, while the second phase extracts the relevant content using the rules obtained in the first phase.

2 A USE CASE: THE WWW WEBPAGE

We present an example that describes a usage scenario of HybEx.

Download. HybEx is an official add-on distributed by Mozilla[1]. It can also be downloaded from http://www.dsic.upv.es/~jsilva/retrieval/Web-HybEx/.

Installation. In most browsers there is an option called "Debug Add-ons" or "Install Add-on from file". Once installed, it automatically adds a new button (H) to the navigation toolbar.

This work has been partially supported by grant PID2019-104735RB-C41 funded by MCIN/AEI/ 10.13039/501100011033, by the *Generalitat Valenciana* under grant Prometeo/2019/098 (DeepTrust), and by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

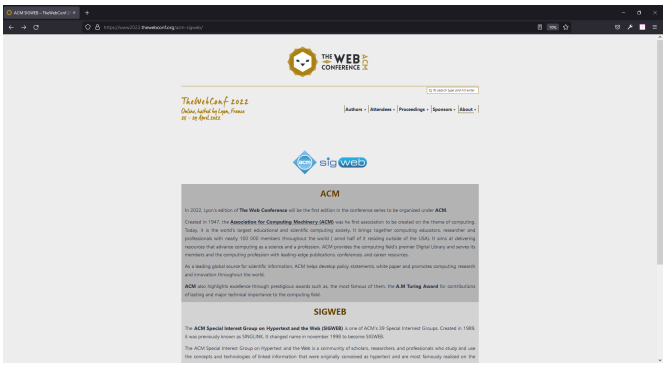
WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524242

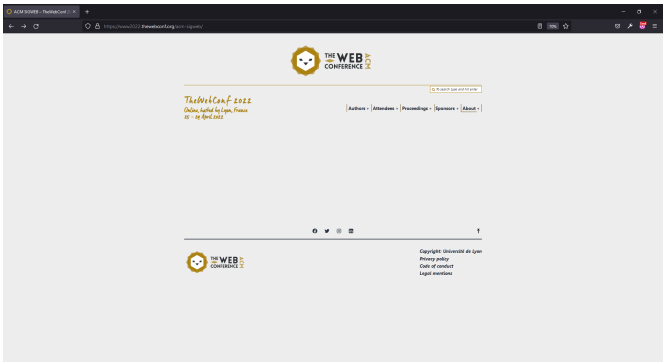
[1] Official Firefox add-ons need to pass a double evaluation process with two different reviewing teams. One for the source code, and other for the functionality.

Use. Browse a webpage normally, and press the add-on button. Then, the template is automatically extracted and displayed.

Example 2.1. *Consider the 'About - ACM SIGWEB' webpage in Figure 1(a) taken from the* WWW 2022 *website. Its template is shared with all webpages of the* WWW 2022 *website. By clicking on the add-on button (H),* HybEx *automatically detects and extracts the template, which is shown in Figure 1(b).*



(a) WWW 2022 webpage



(b) WWW 2022 webpage's template

Figure 1: Template extraction from the WWW 2022 webpage

The extracted template can include any kind of template elements, such as images, HTML/CSS styles, containers, menus, etc. This provides considerable benefits to web programmers and designers, because they can reuse the template by simply copying the HTML and CSS codes. It should be noted in Figure 1(b) that the height of the text container is lower than in Figure 1(a). This is due to the absence of text in that container. The technique has detected the container but, since it has deleted its text, the height of the container is significantly lower. If the add-on button (H) is pressed again, the browser toggles between the extracted template and the original webpage.

3 TOOL ARCHITECTURE

3.1 The hybrid template extractor

HybEx works at the level of DOM. Due to the DOM tree properties, the template of a webpage can be identified with one or several DOM nodes. HybEx consists of a six-step approach (see Figure 2):

(1) First, the HTML of the initial page is transformed into its corresponding DOM tree.
(2) Second, `Page-level ConEx` selects some of the DOM tree's nodes and computes several weights (see [2] for details) standardizing the obtained values. Then, it considers each node as a point in $\mathbb{R}^4$. These points are explored by an algorithm that computes their centroid. Once the centroid is computed, an algorithm builds the set of *candidate nodes* which includes the DOM nodes (points in $\mathbb{R}^4$) located farther to the centroid. Finally, the algorithm selects the DOM node or nodes that correspond to the main content.
(3) The hyperlink analysis algorithm selects a set of webpages from the same website of the initial page.
(4) A set of webpages that form a complete subdigraph is extracted by the complete subdigraph extraction algorithm.
(5) An algorithm converts the HTML of all the webpages in the complete subdigraph into their corresponding DOM trees.
(6) Finally, an algorithm modifies the DOM tree of the initial page removing the main content detected in step 2. Then, each webpage in the complete subdigraph is explored by an algorithm that computes a mapping between its DOM nodes and the DOM nodes of the modified initial page. When it finds that a DOM node in the modified initial page is repeated in any webpage of the complete subdigraph, it updates a counter that reflects the number of times each node is repeated in other webpages. When the number of times a node is repeated is equal to a precomputed threshold, the node belongs to the template. Finally, the tool returns the template.

It should be noted that step 1 is a part of step 5 because step 5 converts the initial page and other webpages from HTML to DOM. In addition, steps 1 and 2; and steps 3, 4, and 5, can be executed in parallel because the output of step 2 is the input of step 6.

3.2 Implementation as a WebExtension

HybEx has been implemented as a WebExtension[2], which is compatible with Mozilla Firefox and Chromium based browsers, among others. Moreover, it is officially distributed by Mozilla as a Firefox add-on.

3.3 Interfaces with other systems

HybEx can be used by humans as well as by automatic systems that need to extract the template of a webpage. For instance, the later case is usual in crawlers and indexers, which often use preprocessing algorithms that identify the template. This produces several benefits to these systems (i.e., reduction of the processing time, storage saving, etc.). Depending on each case, the interface and output produced by the tool is different:

- **Human users:** This is the case described in Section 2. The extracted template is displayed in the browser and can be downloaded as a reusable webpage that keeps the original styles. Moreover, the template is ready to insert content into the containers. For this, the WebExtension changes the visibility property of the DOM nodes that do not belong to the template to `hidden`, i.e., `node.style.visibility = "hidden";` (see Figure 1(b)).

[2]https://addons.mozilla.org/es/firefox/addon/hybrid-template-extractor/

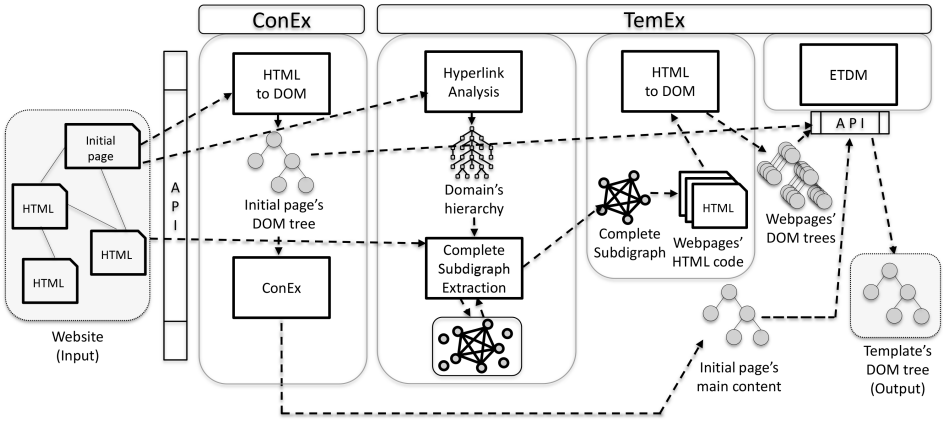


Figure 2: HybEx architecture

In this way, the content can be changed and shown again by changing the `node.style.visibility` property.

- **Non-human users:** Automatic systems can use HybEx with an API to extract the main content, to identify webpage candidates, to infer the template, etc. In this case, the template is returned in several formats, including HTML. One useful output is to return the initial page in HTML format including a new HTML class for those DOM nodes detected as template nodes (`node.className += "template_node";`). This enables the possibility of post-analyzing the initial page, which now includes valuable information about which DOM nodes belong to the template.

4 EMPIRICAL EVALUATION

We conducted a series of experiments using real and online heterogeneous webpages. This allowed us to assess the new template extractor and also to compare it with the original TemEx tool. This was done by measuring the precision, recall, and F1 values (see, e.g., [6] for an explanation about these metrics); and their performance.

The experiments have been done with TeCo (Template detection and Content extraction benchmark suite). We used the 130 benchmarks of version 4.0 (accesible at: http://www.dsic.upv.es/~jsilva/retrieval/teco/).

We divided the 130 benchmarks in a training subset of 80 benchmarks, which we used to train the tool, and an evaluation subset of 50 benchmarks (10 from each category), which we used to assess the tool. Table 1 presents the obtained results. For each benchmark, the first column contains the domain name of the initial pages; column `DOM nodes` indicates the initial page's total number of DOM nodes; column `Template` shows the number of DOM nodes of the template; column `Total Retrieved` contains the number of DOM nodes retrieved as template by the technique; column `Template Retrieved` shows the number of DOM nodes retrieved correctly; column `Recall` contains the quotient of the number of correctly retrieved DOM nodes divided by the number of DOM nodes in the gold standard; column `Precision` contains the quotient of the number of correctly retrieved DOM nodes divided by the total number of retrieved DOM nodes; column `F1` shows the F1 metric, which is computed as $(2 * P * R)/(P + R)$ where P is the precision and R is the recall; finally, column `Runtime` shows the technique's runtime in milliseconds.

The experiments presented in Table 1 reveal a high average F1 value, close to 86%. It should be noted that for two thirds of the benchmarks, the obtained F1 value is higher than 90%. The last row in Table 1 shows the results obtained with the original `TemEx` tool. It can be observed that `HybEx` improves the recall, the precision, and the F1 values obtained by the original `TemEx` template extractor. Regarding the runtime, as expected, the `HybEx` runtime is higher than the runtime of the original `TemEx` template extractor. This is due to the inclusion of the `Page-level ConEx` algorithm as a preprocess for `TemEx`. The overhead introduced is around 250 milliseconds, which is irrelevant in many applications (e.g., if the user is human, it is undetectable).

5 CONCLUSIONS

`TemEx` is a powerful tool that automatically extracts the template of a given webpage. It achieves good results even when its input are heterogeneous webpages and without a predefined fixed template. `HybEx` enhances `TemEx` by integrating a content extraction algorithm (`Page-level ConEx`) as a preprocess. Thus, in a first step `Page-level ConEx` removes the main content from the initial page, and then, `TemEx` extracts the template of the remaining data. In this way, we produce a synergy between both algorithms. Our experiments demonstrate that `HybEx` improves the recall, the precision, and the F1 obtained by the original `TemEx` tool.

REFERENCES

[1] J. Alarte, D. Insa, J. Silva, and S. Tamarit. Temex: The web template extractor. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 155–158, New York, NY, USA, 2015. ACM.
[2] J. Alarte and J. Silva. Page-level main content extraction from heterogeneous webpages. *ACM Trans. Knowl. Discov. Data*, 15(6), jun 2021.
[3] J. Alarte, J. Silva, and S. Tamarit. What web template extractor should i use? a benchmarking and comparison for five template extractors. *ACM Trans. Web*, 13(2), Mar. 2019.
[4] Z. Bar-Yossef and S. Rajagopalan. 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, 2002. ACM.
[5] D. Gibson, K. Punera, and A. Tomkins. The volume and evolution of web page templates. In A. Ellis and T. Hagino, editors, *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*, pages 830–839. ACM, may 2005.

Benchmark	DOM nodes	Template nodes	Total retrieved	Template retrieved	Recall	Precision	F1	Runtime
www.icann.org	485	394	394	393	99,75 %	99,75 %	99,75 %	137
www.museodelprado.es	532	148	165	148	100,00 %	89,70 %	94,57 %	54
www.u-tokyo.ac.jp	602	499	507	499	100,00 %	98,42 %	99,20 %	179
college.harvard.edu	1086	665	265	265	39,85 %	100,00 %	56,99 %	106
www.linuxfoundation.org	578	524	542	524	100,00 %	96,68 %	98,31 %	181
www.einstein.yu.edu	1128	796	869	784	98,49 %	90,22 %	94,17 %	550
www.savethechildren.net	740	679	721	679	100,00 %	94,17 %	97,00 %	1158
parents.berkeley.edu	278	98	96	93	94,90 %	96,88 %	95,88 %	108
www.mensa.es	422	354	373	354	100,00 %	94,91 %	97,39 %	153
www.gip-jci-justice.fr	880	674	755	673	99,85 %	89,14 %	94,19 %	24429
www.diariandorra.ad	1739	610	1095	610	100,00 %	55,71 %	71,56 %	677
www.wishtv.com	2167	1811	1535	1535	84,76 %	100,00 %	91,75 %	11151
www.theday.com	1928	912	1015	911	99,89 %	89,75 %	94,55 %	2391
www.neoteo.com	993	601	600	592	98,50 %	98,67 %	98,58 %	431
www.afp.com	1197	402	375	372	92,54 %	99,20 %	95,75 %	167
www.diariodeburgos.es	568	369	374	365	98,92 %	97,59 %	98,25 %	171
nltimes.nl	550	106	504	106	100,00 %	21,03 %	34,75 %	184
www.journalism.org	755	407	420	406	99,75 %	96,67 %	98,19 %	3042
es.gizmodo.com	575	251	149	149	59,36 %	100,00 %	74,50 %	101
biztechmagazine.com	1892	1053	1794	1053	100,00 %	58,70 %	73,98 %	1092
communities.apple.com	3136	368	2362	368	100,00 %	15,58 %	26,96 %	1182
es.sharelatex.com	1084	870	915	869	99,89 %	94,97 %	97,37 %	913
forums.debian.net	2764	150	405	133	88,67 %	32,84 %	47,93 %	1597
www.meneame.net	750	197	204	197	100,00 %	96,57 %	98,26 %	109
c.mi.com	3473	2932	2600	2590	88,34 %	99,62 %	93,64 %	4124
www.gimpforum.de	1853	449	481	442	98,44 %	91,89 %	95,05 %	1745
alumni.harvard.edu	1975	1756	1838	1756	100,00 %	95,54 %	97,72 %	1218
www.spacetimestudios.com	4871	1371	408	405	29,54 %	99,26 %	45,53 %	203
www.emaildiscussions.com	1086	233	78	78	33,48 %	100,00 %	50,16 %	27
github.com	1233	450	450	450	100,00 %	100,00 %	100,00 %	192
www.folj.com	550	166	413	166	100,00 %	40,19 %	57,34 %	347
johngardnerathome.info	395	176	125	106	60,23 %	84,80 %	70,43 %	1885
www.rosamontero.es	800	83	82	82	98,80 %	100,00 %	99,40 %	44
www.annmalaspina.com	392	182	248	182	100,00 %	73,39 %	84,65 %	81
oneminutelist.com	476	265	308	226	85,28 %	73,38 %	78,88 %	156
artsonline.uwaterloo.ca	410	164	178	164	100,00 %	92,13 %	95,90 %	22
blog.mint.com	822	411	409	399	97,08 %	97,56 %	97,32 %	274
ofdollarsanddata.com	994	343	342	342	99,71 %	100,00 %	99,85 %	1534
benjamincongdon.me	329	55	55	55	100,00 %	100,00 %	100,00 %	18
foodsense.is	330	100	142	100	100,00 %	70,42 %	82,64 %	73
www.arduino.cc	811	475	485	475	100,00 %	97,94 %	98,96 %	264
worryfreelabs.com	497	307	314	307	100,00 %	97,77 %	98,87 %	135
www.tous.com	5079	2987	4999	2987	100,00 %	59,75 %	74,80 %	4896
www.intelligencetest.com	583	320	301	262	81,88 %	87,04 %	84,38 %	194
www.trekbikes.com	5681	1921	1846	1846	96,10 %	100,00 %	98,01 %	4054
www.newprosoft.com	830	151	150	150	99,34 %	100,00 %	99,67 %	66
doodle.com	560	478	483	477	99,79 %	98,76 %	99,27 %	526
us.pandora.net	6195	1977	4572	1965	99,39 %	42,98 %	60,01 %	2717
www.euroholds.com	4869	790	669	669	84,68 %	100,00 %	91,70 %	822
naranjascarcaixent.com	290	148	151	147	99,32 %	97,35 %	98,33 %	86
Avg. HybEx	1444,26	632,56	771,22	578,12	92,13 %	86,14 %	85,65 %	1519,32
Avg. TemEx	1444,26	632,56	757,14	575,50	91,00 %	85,86 %	84,81 %	1264,68

Table 1: Results of the experimental evaluation

[6] T. Gottron. Evaluating content extraction on html documents. In *Proceedings of the 2nd International Conference on Internet Technologies and Applications (ITA'07)*, pages 123–132, 2007.

[7] J. Leonhardt, A. Anand, and M. Khosla. *Boilerplate Removal Using a Neural Sequence Labeling Model*, page 226–229. Association for Computing Machinery, New York, NY, USA, 2020.

[8] S. S. Modi and S. B. Jagtap. Multimodal web content mining to filter non-learning sites using nlp. In A. Pandian, T. Senjyu, S. M. S. Islam, and H. Wang, editors, *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2018)*, pages 23–30, Cham, 2020. Springer International Publishing.

[9] D. d. C. Reis, P. B. Golgher, A. S. Silva, and A. H. F. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*, pages 502–511, New York, NY, USA, 2004. ACM.

[10] D. Song, F. Sun, and L. Liao. A hybrid approach for content extraction with text density and visual importance of dom nodes. *Knowledge and Information Systems*, 42(1):75–96, Jan 2015.

[11] N. Utiu and V.-S. Ionescu. Learning web content extraction with dom features. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 5–11, 2018.

[12] E. Uzun, H. V. Agun, and T. Yerlikaya. A hybrid approach for extracting informative content from web pages. *Information Processing and Management*, 49(4):928 – 944, 2013.

[13] 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)*, pages 258–267, New York, NY, USA, 2006. ACM.

[14] T. Vogels, O. Ganea, and C. Eickhoff. Web2text: Deep structured boilerplate removal. *CoRR*, abs/1801.02607, 2018.

[15] L. Yi, B. Liu, and X. Li. Eliminating noisy information in web pages for data mining. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD'03)*, pages 296–305, New York, NY, USA, 2003. ACM.

[16] H. Zhang and J. Wang. Boilerplate detection via semantic classification of textblocks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[17] S. Zhang, J. Wu, and K. Yang. A webpage segmentation method based on node information entropy of DOM tree. *Journal of Physics: Conference Series*, 1624(3):032023, oct 2020.