TeMex: The Web Template Extractor*

Julián Alarte & David Insa & Josep Silva
Departamento de Sistemas Informáticos y
Computación
Universitat Politècnica de València
Camino de Vera s/n
E-46022 Valencia, Spain.
{jalarte,dinsa,jsilva}@dsic.upv.es

Salvador Tamarit
Babel Research Group
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte (Madrid), Spain.
stamarit@babel.ls.fi.upm.es

ABSTRACT

This paper presents and describes TeMex, a site-level web template extractor. TeMex is fully automatic, and it can work with online webpages without any preprocessing stage (no information about the template or the associated webpages is needed) and, more importantly, it does not need a predefined set of webpages to perform the analysis. TeMex only needs a URL. Contrarily to previous approaches, it includes a mechanism to identify webpage candidates that share the same template. This mechanism increases both recall and precision, and it also reduces the amount of webpages loaded and processed. We describe the tool and its internal architecture, and we present the results of its empirical evaluation.

1. INTRODUCTION

This article presents TeMex, a tool able to automatically extract the template of a website. Template extraction is a hot topic with multiple applications, and thus, there exist several other approaches for template extraction (see, e.g., [4, 8, 10]). However, our new technique produces the best recall and precision in the state of the art with a remarkably improved performance. Moreover, thanks to a new algorithm, our tool needs to explore significantly less webpages to detect the template. TeMex is ready to be used by other systems, such as crawlers, and also by human users, because it is distributed as a (official) Firefox add-on.

1.1 Motivation

A web template (in the following just template) is a prepared HTML page where formatting is already implemented and visual components are ready to insert content into them.


Templates are used as a basis for composing new webpages that share a common look and feel. This is good for web development because many tasks can be automated thanks to the reuse of components. In fact, many websites are maintained automatically by code generators that generate webpages using templates. Templates are also good for users, which can benefit from intuitive and uniform designs with a common vocabulary of colored and formatted visual elements.

Templates are also important for crawlers and indexers, because they usually judge the relevance of a webpage according to the frequency and distribution of terms and hyperlinks. Since templates contain a considerable number of common terms and hyperlinks that are replicated in a large number of webpages, relevance may turn out to be inaccurate, leading to incorrect results (see, e.g., [4, 12, 10]). Moreover, in general, templates do not contain relevant content, they usually contain one or more regions [4] where the main content must be inserted. Therefore, detecting templates helps indexers to identify the main content of the webpage. Gibson et al. [6] determined that templates represent between 40% and 50% of data on the Web and that around 30% of the visible terms and hyperlinks appear in templates. This justifies the importance of template removal [12, 10] for web mining and search.

2. CASE OF USE

In this section we show an example of usage of TeMex. TeMex is distributed as an official Firefox add-on, and it is extremely easy to download, install and use.

Download. TeMex can be downloaded from the Firefox add-ons repository. It can also be downloaded from:
http://www.dsic.upv.es/~jsilva/retrieval/templates/
It comes as a XPI file that packages the whole add-on.

Installation. Just drag and drop the XPI file on the main Firefox window. It automatically adds a new button (the TeMex button: T) to the Firefox navigation toolbar.

Use. Navigate normally to a webpage, and press the TeMex button. The template is automatically shown.

EXAMPLE 2.1. *Consider the* WWW 2015 *main webpage shown in Figure 1(a). This webpage shares a template with all webpages of the* WWW 2015 *website. By clicking the* TeMex *button, the tool automatically detects its template. Figure 1(b) shows the template extracted by* TeMex.

Observe that the template produced by TeMex *includes images, styles, HTML containers, and all web components that belong to the template. Therefore, it is ready to insert code, and can be reused by web engineers. Note also that* TeMex *identifies the texts* "Welcome to the 24th..." *and* "published in Facebook and Twitter" *as part of the template. This is*
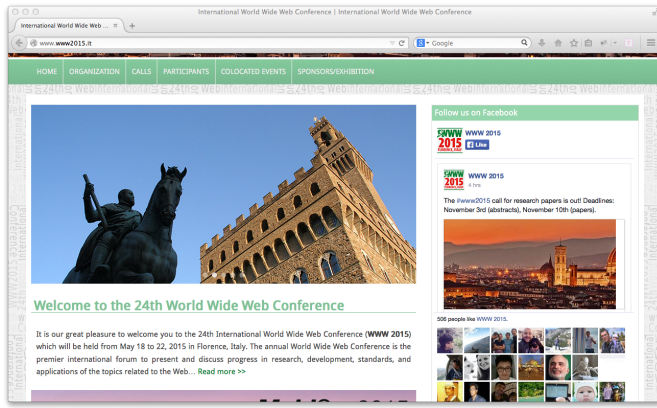
(a) WWW 2015 main webpage
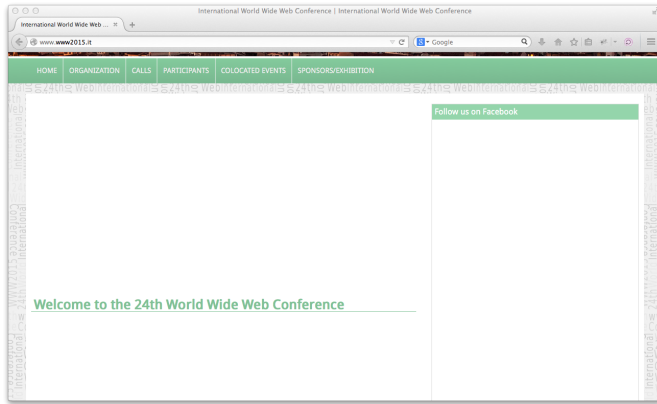
(b) WWW 2015 template

Figure 1: WWW 2015 main webpage

not an error. This is an indication of `TeMex` *that the template contains a text in this position. Of course, this text can change between the different webpages, but the text appears in all of them, and thus, it is part of the template. The concrete text displayed by* `TeMex` *is the text appearing in the original webpage.*

After extracting the template, the user can change the view swapping between the template and the webpage (and vice-versa) just pressing again the `TeMex` *button.*

3. INTERNAL ARCHITECTURE

3.1 The template extraction technique

`TeMex` implements various novel algorithms that make it more precise and efficient than other similar tools. Essentially, the internal template extraction technique works in three phases.

1. **Detection of webpage candidates that share the template:** The input of `TeMex` is a webpage (in the following *key page*) whose template is not necessarily shared with the other webpages in the website. Therefore, the first step is to explore the website searching for a set of webpages that share the template with the key page. To identify these webpages we use a novel technique [2] that detects the menu of the key page by calculating a complete sub-digraph (CS) in the website topology. The webpages pointed out by the menu are normally the homepage of different sections of the website; and these webpages often share the same template.

With these ideas, and using an hyperlink analysis, to select the best CS, and the best components of the CS, the technique can select the best candidates with very few webpage loads (a mean of 5.3).

2. **Comparison of the key page with the set of webpage candidates:** The template is extracted by comparing the key page with the collected webpages. This comparison is done at the level of DOM, i.e., the template is a subset of the DOM nodes of the key page (those shared with *a majority of* the other webpages). The comparison is done using a graph theory formalism called *Equal Top-Down Mapping* (ETDM) that establishes a relation between two given DOM trees. Roughly, the algorithm that compares DOM trees uses a voting system that determines that a node belongs to the template if it appears in more DOM trees than a precomputed threshold.

3. **Filtering of the key page:** The template is reconstructed using a DOM tree copy of the key page. An algorithm processes it removing those parts that do not belong to the template. In this way the technique can output a well-formed webpage as the template.

3.2 Integration into Firefox

`TeMex` is distributed as a Firefox add-on. Firefox is one of the most powerful and widely used browsers, and it is free and open source. This allows us to access and manipulate the internal data structures used to handle webpages as DOM trees. Moreover, Firefox offers important architectonical advantages because add-ons have direct access to the internal Firefox API, and also because the design of the add-ons layer clearly separates functionality and GUI with specific languages and facilities for them. In particular, Firefox toolbars use XUL, an XML based language, to design the GUI. And they use Javascript, to implement the behavior and event-handling. The current version of `TeMex` (`TeMex` `1.5`) contains 2594 LOC.

Between the multiple kinds of firefox extensions, `TeMex` is an *overlay extension* [1]. Therefore, it uses XUL overlays to specify the interface, and APIs available to privileged code such as tabbrowser and Javascript modules to interact with the application and content.

Figure 2 shows our Firefox add-on architecture. It uses four main modules to implement two functionalities: "Extract Template" (executed the first time that the `TeMex` button is pressed) and "Toggle View" (executed the second and successive times that the `TeMex` button is pressed). "Extract Template" (1) uses a module to analyze the key page and explore its links to obtain the topology of the website. (2) This topology is used by another module to identify a CS in the domain graph. Then, (3) `TeMex` compares the DOM trees of the CS webpages with the key page's DOM tree to finally obtain the website template. "Toggle View" uses a single module to swap the webpage displayed (original ↔ template) by loading the appropriate DOM tree.

3.3 Interfaces with other systems

`TeMex` can be used by human users and by other systems that need to extract the template of a webpage (e.g., as a preprocessing stage). The later case is common in crawlers, which enhance searching and indexing processes thanks to the identification of the template. The interface and output produced by `TeMex` is different in each case:

- **Human users:** For human users `TeMex` implements a GUI. In this case, the template is displayed as a reusable webpage where containers are ready to insert content
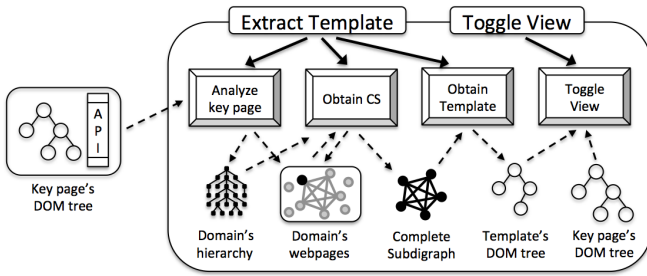
Figure 2: Firefox add-on architecture

and styles are kept. In order to keep the same structure of the key page, the template is displayed by changing the visibility property of those non-template DOM nodes to hidden, i.e., `node.style.visibility = "hidden";` (see Figure 1(b)).

- **Non-human users:** In this case, other systems can use `TeMex` with an interface to detect webpage candidates, to extract the template, etc. The template is returned in different formats, including HTML. It is also possible to output the HTML of the key page where those components that belong to the template have been included into an HTML class (`node.className += "template_node";`). In this way, it is possible to post-analyze the key page with explicit information about what parts belong to the template.

4. EMPIRICAL EVALUATION

We conducted several experiments with real, online webpages to provide a measure of the average performance regarding recall, precision and the F1 measure (see, e.g., [7] for a discussion on these metrics). Initially, we wanted to use a public standard collection of benchmarks, but we did not find any public dataset for template extraction. In particular, we could not use the standard CleanEval suite [5] of content extraction benchmarks, because it contains a gold standard prepared for content extraction (each part of the webpages is labelled as *main-content* or *non-content*), but it is not prepared for template extraction. Then, we tried to use the same benchmark set as the authors of other template extraction papers. However, due to privacy restrictions, copyright, or unavailability[1] of the benchmarks we could not use a previous dataset. Therefore, we decided to create a new suite of benchmarks. We created TECO, a new publicly accessible dataset with an automatizable gold standard.

4.1 The TECO benchmark suite

TECO (TEmplate detection and COntent extraction benchmark suite) is a dataset of 70 real and heterogeneous webpages with different layouts and page structures, including different languages to allow the testing of language independence features. Each benchmark in the suite is composed of a set of webpages: the key page and all webpages that can be reached from the key page with a maximum depth of three clicks. The webpages and all their resources (images, media, CSS, Javascript, etc.) have been localized so that all links reference their local copy to ensure independency of the benchmark with respect to the evolution of the websites. For each benchmark, we have manually determined its main content and its template, and we have labelled every single

---

[1]Some authors answered that their benchmarks were not stored for future use, or that they did not save the gold standard.

element with a *mainContent*, *template*, *notContent*, and/or *notTemplate* class. Therefore, the suite is useful for both content and template extraction. With TECO, researchers can evaluate or compare their technique very easily thanks to the labeling and also because the suite also includes scripts to automatize the analysis of webpages. TECO is open and free, and it is available at:
`http://www.dsic.upv.es/~jsilva/retrieval/teco/`
A detailed description of the suite can be found in [3].

4.2 Experiments

Table 1 summarizes the results of the performed experiments. The first column contains the key pages' URL of the evaluated websites (we used 40 benchmarks for training and 30 benchmarks for evaluation). For each benchmark, column `DOM nodes` shows the number of nodes of the key page's DOM tree; column `Template nodes` shows the number of nodes of the gold standard template; column `Total Retrieved` shows the number of nodes that were identified by the tool as the template; column `Template Retrieved` shows the number of nodes retrieved that belong to the gold standard template; column `Recall` shows the number of correctly retrieved nodes divided by the number of nodes in the gold standard; column `Precision` shows the number of correctly retrieved nodes divided by the number of retrieved nodes; finally, column `F1` shows the F1 metric that is computed as $(2 * P * R)/(P + R)$ being P the precision and R the recall.

Experiments reveal a high average precision and recall: more than 85% in both cases. This was computed by calculating a CS of size 3. We observed that other techniques such as [4, 10, 7, 9, 11] obtain good values of F1 in certain webpages, but they are manually feed with collections of webpages that share the same template. With this conditions, our tool produces an F1 close to 95% in most of the cases. For instance, in [10] the authors get an F1 between 85% and 95% with collections of 24 webpages where all of them implement the same template (e.g., all of them are product descriptions). In our experiments, in contrast, our tool inputs heterogeneous webpages that not necessarily implement the same template. Moreover, our tool only needs an average of 5.3 webpages loaded to get the results in Table 1.

The F1 measure reported in the bibliography is different for each paper. Some of them measure the number of words correctly retrieved [10, 9]. This can be rather imprecise, because it ignores the structure (e.g., `div`, `table`...) retrieved. In our experiments we measured DOM nodes, which is the smallest granularity measure, and it takes into account the text and structure retrieved. We wanted to compare all techniques with the same benchmarks and with the same measure, but we could not access to the implementation of the tools even if they were reported as free. It is surprising, and quite disappointing, to see how few systems are open-source, or even otherwise (freely) available. In many papers, it is stated that a prototype was developed but we were not able to find the tool. To solve this, we are currently reimplementing the main template extraction systems in literature with the same language. This will allow us to compare them with the same benchmarks, measures and criteria. Some of them are already available as open-source at:
`http://www.dsic.upv.es/~jsilva/retrieval/templates/`

5. CONCLUSIONS

`TeMex` is a tool able to automatically extract the template of a given webpage. It implements new analyses and techniques to detect the menu of the website in an efficient way. The webpages pointed out by the menu are used to compare their

Benchmark	DOM nodes	Template nodes	Total retrieved	Template retrieved	Recall	Precision	F1
www.accountkiller.com/en/	501 nodes	222 nodes	222 nodes	222 nodes	100 %	100 %	100 %
parents.berkeley.edu/advice/	282 nodes	98 nodes	95 nodes	95 nodes	96.94 %	100 %	98.45 %
switzerland.isyours.com/e/	571 nodes	565 nodes	519 nodes	519 nodes	91.86 %	100 %	95.76 %
today.java.net/pub/	695 nodes	341 nodes	322 nodes	305 nodes	89.44 %	95 %	96.20 %
en.proverbia.net/citastema.asp	372 nodes	126 nodes	131 nodes	126 nodes	100 %	96.18 %	98.05 %
www.brighthand.com/news/	1116 nodes	1116 nodes	815 nodes	815 nodes	73.03 %	100 %	84.41 %
www.hazards.org/rehab/	134 nodes	134 nodes	118 nodes	118 nodes	88.06 %	100 %	93.65 %
www.moderncreative.com/services/	364 nodes	318 nodes	275 nodes	229 nodes	72.01 %	83.27 %	77.23 %
www.netcomuk.co.uk/~rwevans1/	671 nodes	149 nodes	93 nodes	93 nodes	62.42 %	100 %	76.86 %
www.prc.org/resources_student.html	528 nodes	199 nodes	177 nodes	177 nodes	88.94 %	100 %	94.15 %
www.robyncarr.com/qa.html	292 nodes	92 nodes	40 nodes	40 nodes	43.48 %	100 %	60.61 %
www.strangehorizons.com/2004/	634 nodes	149 nodes	154 nodes	149 nodes	100 %	96.75 %	98.35 %
www.facts-about-japan.com/	504 nodes	431 nodes	467 nodes	431 nodes	100 %	92.29 %	95.99 %
www.userfriendly.org/community/	244 nodes	105 nodes	4 nodes	4 nodes	3.81 %	100 %	7,34 %
www.armscontrol.org/act/	836 nodes	512 nodes	334 nodes	298 nodes	58.20 %	89.22 %	70.45 %
melizzard.typepad.com/	565 nodes	265 nodes	281 nodes	265 nodes	100 %	94.31 %	97.07 %
www.uniteddesign.com/	232 nodes	99 nodes	26 nodes	26 nodes	26.26 %	100 %	41.6 %
www.rocklists.com/91x-1983.html	765 nodes	533 nodes	583 nodes	533 nodes	100 %	91.42 %	95.52 %
pages.jh.edu/~jhumag/	393 nodes	94 nodes	89 nodes	89 nodes	94.68 %	100 %	97.27 %
www.intelligencetest.com/mindgames/	366 nodes	284 nodes	281 nodes	281 nodes	98.94 %	100 %	99.47 %
doodle.com/online-calendar.html	572 nodes	490 nodes	496 nodes	490 nodes	100 %	98.79 %	99.39 %
worryfreelabs.com/jobs/index.html	424 nodes	321 nodes	321 nodes	321 nodes	100 %	100 %	100 %
ernstfamily.ch/jonathan/	219 nodes	104 nodes	216 nodes	104 nodes	100 %	48.15 %	65 %
golang.org/doc/install/gccgo.html	717 nodes	78 nodes	78 nodes	78 nodes	100 %	100 %	100 %
www.newprosoft.com/	832 nodes	151 nodes	148 nodes	148 nodes	98.01 %	100 %	99.00 %
www.folj.com/puzzles/	559 nodes	175 nodes	452 nodes	171 nodes	97.71 %	37.83 %	54.55 %
www.alt-codes.net/	772 nodes	237 nodes	237 nodes	237 nodes	100 %	100 %	100 %
cluster013.ovh.net/~polcapro/	491 nodes	238 nodes	238 nodes	238 nodes	100 %	100 %	100 %
www.craftcoffee.com/	610 nodes	345 nodes	348 nodes	345 nodes	100 %	99.14 %	99.57 %
oneminutelist.com/	490 nodes	273 nodes	309 nodes	233 nodes	85.35 %	75.40 %	80.07 %
Average	525 nodes	274.8 nodes	262.3 nodes	239.33 nodes	85.64 %	93.25 %	85.73 %

Table 1: Results of the experimental evaluation

DOM trees in order to identify what parts form the template. Our experiments demonstrate that the tool only needs to load around 5 webpages from the website to extract the template, and it produces a 85.73% of F1.

6. REFERENCES

[1] Overlay extension. Available at URL: https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions, 2005.

[2] J. Alarte, D. Insa, J. Silva, and S. Tamarit. Automatic Detection of Webpages that Share the Same Web Template. In M. H. ter Beek and A. Ravara, editors, *Proceedings of the 10th International Workshop on Automated Specification and Verification of Web Systems (WWV 14)*, volume 163 of *Electronic Proceedings in Theoretical Computer Science*, pages 2–15. Open Publishing Association, July 2014.

[3] J. Alarte, D. Insa, J. Silva, and S. Tamarit. A Benchmark Suite for Template Detection and Content Extraction. *CoRR*, abs/1409.6182, 2014.

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

[6] 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.

[7] T. Gottron. Evaluating content extraction on HTML documents. In V. Grout, D. Oram, and R. Picking, editors, *Proceedings of the 2nd International Conference on Internet Technologies and Applications (ITA'07)*, pages 123–132. National Assembly for Wales, sep 2007.

[8] 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.

[9] K. Vieira, A. L. da Costa Carvalho, K. Berlt, E. S. de Moura, A. S. da Silva, and J. Freire. On finding templates on web collections. *World Wide Web*, 12(2):171–211, 2009.

[10] 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.

[11] T. Weninger, W. Henry Hsu, and J. Han. CETR: Content Extraction via Tag Ratios. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*, pages 971–980. ACM, apr 2010.

[12] 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.