



Impact of template removal on web search*

Impacto de remoção de templates em busca na web

Kaio Wagner¹
Edleno Silva de Moura²
David Fernandes³
Marco Cristo⁴
Altigran Soares da Silva⁵

Resumo

Trabalhos na literatura têm indicado que templates de páginas Web representam ruído em coleções Web e advogam que a simples remoção de tais templates contribui para a melhoria da qualidade das respostas fornecidas por sistemas de busca. Neste artigo, nós estudamos o impacto da remoção de templates em dois cenários distintos: (a) coleções de escala Web, ou seja, consistindo de diversos sítios diferentes; (b) intrasites, ou seja, um sítio isolado, onde se pretende fazer buscas. Nosso trabalho é o primeiro na literatura a estudar o impacto de remoção de templates em coleções de escala Web. O estudo foi realizado utilizando um método automático de detecção de templates que propusemos em um trabalho anterior. Como contribuições, (a) apresentamos estatísticas sobre a aplicação deste método de detecção na coleção de referência GOV2, uma coleção de escala Web; (b) comparamos o desempenho do detector de templates com um processo de detecção manual e (c) observamos que, em ambos os cenários estudados, a remoção de templates não melhorou a qualidade dos resultados obtidos pelo sistema de busca mas foi capaz de reduzir significativamente o tamanho dos índices usados.

Palavras-chave: Remoção de templates. Detecção de templates. Indexação.

*Invited paper.

¹Universidade Federal do Amazonas, AM, Brazil – kaio@dcc.ufam.edu.br.

²Universidade Federal do Amazonas, AM, Brazil – edleno@dcc.ufam.edu.br.

³Universidade Federal do Amazonas, AM, Brazil – david@dcc.ufam.edu.br.

⁴Universidade Federal do Amazonas, AM, Brazil – marco.cristo@dcc.ufam.edu.br.

⁵Universidade Federal do Amazonas, AM, Brazil – alti@dcc.ufam.edu.br.

Abstract

Previous work in literature has indicated that template of web pages represent noisy information in web collections, and advocate that the simple removal of template result in improvements in the quality of results provided by Web search systems. In this paper, we study the impact of template removal in two distinct scenarios: large scale web search collections, which consist of several distinct websites, and intrasite web collections involving searches inside of web sites. Our work is the first in literature to study the impact of template removal to search systems in large scale Web collections. The study was carried out using an automatic template detection method previously proposed. As contributions, we present statistics about the application of this automatic template detection method to the well known GOV2 reference collection, a large scale Web collection. We also present experiments comparing the amount of template detected by our automatic method to the ones obtained when humans select templates. And finally, experiments which indicate that, in both experimented scenarios, template removal does not improve the quality of results provided by search systems, but can play the role of an effective loss compression method by reducing the size of their indexes.

Keywords: Template removal. Template detection. Indexing.

1 INTRODUCTION

Templates are blocks of HTML code shared by a set of pages in the same website. They are widely used by content providers to enhance the uniformity of the layout and navigation on their websites. In this paper, we evaluate the impact of template removal on quality of results in search engines.

Many authors have argued that content on templates represent noisy information and should be disregarded to avoid deteriorating search precision. However, we have found no work in literature dedicated to properly assert the effectiveness of such strategy. In this work, we examine this hypothesis through experiments in two scenarios: (i) large-scale Web scenario, and (ii) intrasite scenario. For the large-scale scenario, we performed experiments with the TREC GOV2 collection, which contains more than 25 millions pages crawled from “.gov” domain. For the intrasite scenario, we performed experiments on three websites with heterogeneous structures, and on a collection containing pages of nine popular blogs.

Despite the general belief that template removal may improve the quality of search results (VIEIRA; PINTO, 2006; YI; LIU; LI, 2003), our experiments suggest the improvements are not consistent across various sites, and that template removal in general does not cause improvements on quality of results provided by search systems. Indeed, in three of the test collections used in the intrasite scenario, the results obtained with template removal were equivalent to results without template removal, for the queries we experimented with. Further, we notice a slightly negative impact in the quality of the results in the large scale scenario after removing templates.

The study conducted in this article presents two major contributions to web search researchers. First, it shows through empirical experimentation that, contrary to what is advocated in several previous work, templateremoval does not cause significant improvement in quality of search results. Second, the results obtained indicate that the information extracted from templates may be useful to specific types of queries, and thus template information might be used as a separate source of relevance evidence, instead of noisy information to be removed from the index of search systems.

2 RELATED WORK

Many authors have hypothesized that templates are typically not related to the main content of the pages and might hurt search quality (BARYOSSEF; RAJAGOPALAN, 2002; VIEIRA; PINTO, 2006; YI; LIU; LI, 2003). Such an idea has motivated the proposal of algorithms to detect and remove templates which led to gains according to preliminary evaluations (BARYOSSEF; RAJAGOPALAN, 2002; VIEIRA; PINTO, 2006; YI; LIU; LI, 2003). How-

ever, no systematic study was conducted by those authors to support the original hypothesis. Other works (MA et al., 2003; CHEN; YE; LI, 2006) have also been able to improve search quality by removing templates, although with much smaller gains on quality, as detailed in the following paragraph.

In (MA et al., 2003), it was proposed a straightforward method to identify and remove templates from Web pages. First, the pages are segmented according to its <TABLE> tags and then the most frequent segments in the collection are treated as templates. The authors evaluated their method by removing the templates found in a page collection extracted from CNN. They concluded that (a) the resulting index was about 21% smaller than the original one and (b) the precision of the results was increased for queries which previously yielded non relevant results when submitted to the original index. A shortcoming of this study was the artificial nature of the queries used in the experiments. They were pre-selected such that only single term queries were used and all the query terms were present in the template vocabulary. From out the 280 queries submitted, 39 received better answers than in the original collection whereas only 5 performed worse. No difference was noted for the remaining queries. To work efficiently, the proposed method has to avoid replicated pages which make it unfeasible to be applied to a large scale page collection, a typical search engine scenario.

In (CHEN; YE; LI, 2006), it was proposed an alternate approach for detecting templates in large scale page collections. Their method divides the pages into blocks and groups them according to their location in the page. Similar groups are then identified as part of the template of the pages. To evaluate their approach, they conducted a small experiment which consisted in applying its method to detect templates in eight popular sites and then submitted twenty queries to one of these sites. They found a small increase in the precision of the answers obtained.

Other works in literature have observed no advantage in template removal besides some reduction in the index size (VIEIRA; SILVA, 2008; WAN; THOMAS; ROWLANDS, 2009; FERNANDES et al., 2007). For instance, Vieira and Silva (2008) carried out experiments to evaluate the impact of three template detection algorithms on a new page collection extracted from a Brazilian Web portal¹. They evaluated methods SST (YI; LIU; LI, 2003), RBM-TD (VIEIRA et al., 2009), and RTDM-TD (VIEIRA et al., 2009). By using RBM-TD and RTDM-TD, they observed a significant reduction on the index with no impact on search quality. As for SST, significant index reductions were accompanied by a loss in the quality of the answers.

In (WAN; THOMAS; ROWLANDS, 2009), the detection and removal template was treated as a longest common subsequence problem, for which they proposed an efficient algorithm. By performing experiments with the WT10g collection and with a page collection from the media domain, they obtained reductions in the resulting indices ranging from 9 to 54%. However, no increase in the quality of the answers was observed.

In (FERNANDES et al., 2007), the templates were manually removed from two Web

¹<http://ultimosegundo.ig.com.br>

collections to avoid eventual mistakes inherent to the completely automatic detection algorithms. As the previous two works, they also verified that the simple removal of templates did not lead to an increase in the quality of the answers provided by the search engine.

None of these works have evaluated the hypothesis that template removal should be advantageous to search engines through complete and satisfactory experiments. As a consequence, it is still not clear if the task of template removal pays off. Unlike previous work, we here provide a study on the validity of that hypothesis. In particular, this is the first work to study the impact of template removal in a scenario which is closer to a real Web search engine, when compared to previous work, using a large number of sites and a reference query set.

3 TEMPLATES

As defined before, templates are portions of HTML code and textual content shared by different pages in the same website. Previous research works demonstrated that the templates represent a substantial portion of the content available in the Web (WAN; THOMAS; ROWLANDS, 2009; GIBSON; PUNERA; TOMKINS, 2005). The reasons for this wide adoption of templates are numerous. The templates provide a consistent look across the pages of a website, increasing its usability and navigability. Further, the adoption of templates increases the speed of creation of Web sites and reduces the effort of web designers.

Recent advances in Web site engineering also contributed to this wide adoption of templates. For instance, many web design software, such as Adobe Dreamweaver and Microsoft Expression Web, provides capabilities for creating pages using a small set of fixed templates. Also, in large websites, steady templates and textual contents can be combined by the server-side technologies to produce the web documents of a site. This separation between content and presentation is one of most often cited advantage of using Web templates.

Figure 1 depicts a page of the CNET News² website. In this Figure, the dotted lines highlight the segments belonging to the template of the page, which include a navigation bar, a search box, advertising, most popular news, etc. Since the terms of these types of segments are typically not related to the main content of the pages, many authors have classified these pieces as noise. However, the templates can also contain more informative segments. To illustrate, a set of related pages can share a segment that inform the users about their common purpose. For instance, the pages of science fiction books of a virtual library can share a segment containing the terms of their common genre. Further, some segments of the templates may contain generic information about the whole Web site. For instance, the pages of a virtual library can share a segment that make clear the purpose of the site, showing terms as books, store, etc.

²news.cnet.com

Figure 1 – A Web page and their template segments.



4 TEMPLATE DETECTION ALGORITHMS

Because the potential negative impact on the information retrieval systems, several authors have proposed methods to use the structure of Web pages as a means of identifying noisy content. For instance, the authors in (BARYOSSEF; RAJAGOPALAN, 2002) proposed mechanisms for partitioning Web pages into segments and selecting some of these segments as candidate templates using some properties of their textual content. However, the method proposed to identify segments is only based on the number of links of each HTML node, and thus may not reflect the layout of the pages. Poor segmentation will cause the failure of template detection.

In (YI; LIU; LI, 2003), it was proposed an alternative approach whose main goal is to find common noisy segments of a set of pages. Their approach is based on a tree structure, called style tree, that summarizes all the presentation styles found in the pages of a Web site. It is built by scanning a set of pages and adding to SST every distinct DOM node found. To classify each node of the style tree into useful or noise content, the authors propose some entropy based indicators. A drawback of this method is the number of pages needed to obtain reliable statistic data. Further, the SST can require large amounts of memory when applied to large Web sites.

In (SONG et al., 2004) was adopted a visual-based segment division proposed by (CAI et al., 2003) to obtain segment information. They used a machine learning approach to compute

an importance rank of the segments found. Spatial features, such as the position and size of the segments, and content features, such as the number of images and links, are extracted from each segment. Then, two learning algorithms, neural networks and Support Vector Machines (SVM), are used to rank the segments according to their importance. Segments assigned with a low weight were considered as belonging to the template of the pages.

4.1 The Algorithm MTD

The template detection methods described in previous section identify templates in collections which have just one template, i.e., homogeneous collections. An example of homogeneous collection is the set of pages extracted from a News site which uses the same template for all its articles regardless their subjects (Politics, Sports, Economy, etc). However, many collections, such as most of the ones adopted in this work, were crawled from multiple sites. As such, they are heterogeneous as they have many templates. Note that, even when the collection is extracted from just one site, it can be heterogeneous.

Since collections used in a Web search engine scenario are heterogeneous, in this study we use a method able to detect multiple templates. In particular, we adopted method Multiple Template Detection (MTD) (VIEIRA et al., 2009).

The MTD algorithm starts by partitioning the collection into groups of pages that probably share the same template. It then uses an algorithm able to detect templates in homogeneous collections to identify the template of each partition. For this task, it uses the Restricted Bottom-up Mapping (RBM-TD) proposed in (VIEIRA et al., 2009), which correctly identifies templates by performing a bottom-up matching between the DOM trees of the pages being considered, an operation that is linear in the worst case, as we will see in next section.

To sustain a high scalability, MTD uses two distinct functions as follows. Given a page p , the first function selects a small set of candidate partitions according to the DOM tree complete paths that p shares with the pages of the partition. A second function attempts to find a same template in p and in the pages of the candidate partitions. Thus, the first function filters out the pages to be processed by the second function, as we can see in Algorithm 1.

Algorithm 1: Algorithm MTD

```

1: MTD(DB)
2: Input: DB / * Set of pages * /
3: Output: C / * Set of page partitions in DB * /
4: C  $\leftarrow \{\}$ 
5: for each  $p \in \text{DB}$  do
6:   fits  $\leftarrow$  false
7:   CL = { $c_i \in C \mid \text{PathFitness}(p, c_i) >_{\text{path}}$ }
8:   sort CL according to PathFitness in decreasing order
9:   for each  $c_i \in \text{CL}$  do
10:    if fits = false then
11:      template  $\leftarrow \text{DetectSingleTemplate}(\text{temp}_i, p)$ ;
12:      if |template|  $>_{\text{tree}}$  then
13:         $c_i \leftarrow c_i \cup p$ 
14:         $\text{temp}_i \leftarrow$  true
15:        fits  $\leftarrow$  true
16:      end if
17:    end if
18:  end for
19:  if fits = false then
20:    Create new cluster  $c_{\text{new}}$ 
21:     $c_{\text{new}} \leftarrow \cup \{p\}$ 
22:     $\text{temp}_i \leftarrow \{p\}$ 
23:  end if
24: end for

```

MTD groups pages according to their templates while detect them at same time. The grouping is performed by matching, for each page p , the DOM tree paths of p with those in the templates of the partitions previously found. MTD selects only groups which share at least path paths with p , which results in a small list of candidate partitions (CL) for p (see Algorithm 1, line 07, where $\text{PathFitness}(p, c_i)$ denotes the number of paths that p shares with partition c_i).

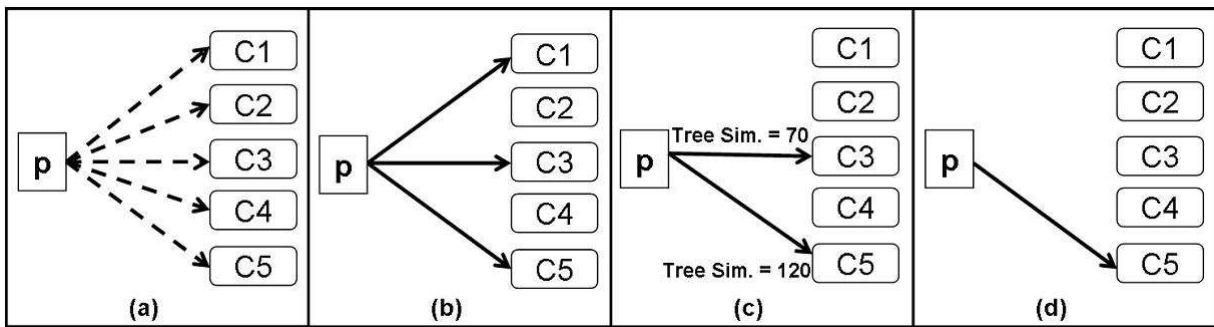
The Function $\text{DetectSingleTemplate}(\text{temp}_i, p)$ (line 8) implements any single template detection method which is able to find a common subtree between temp_i , the template of current group c_i , and page p . As previously mentioned, in particular, our implementation of MTD uses RBM-TD as the $\text{DetectSingleTemplate}$ function. Page p can be included in c_i only if the number of nodes in the template is greater than tree (line 9). In that case, p is added to the partition and the template subtree is taken as the new template of partition c_i (line 10-11). On the other hand, a new partition c_{new} is created, containing only p , which is also taken as the template of the partition (lines 17-20).

Value path is a tradeoff between computational cost and grouping quality. The higher is the path value, the smaller is the number of candidate partitions to verify. As a consequence, by using a high path value, the algorithm can miss good partition candidates whereas, by using a small value, it spends more time verifying a larger list of candidates.

The Figure 2 illustrates how MTD works. First, page p can be added to any existing

partition (Figure 2(a)). From these partitions, MTD filters out the ones whose pages do not share paths with p (Figure 2(b)). Then, a new filtering out process is carried out such that only partitions with more than tree nodes in common with p are considered (Figure 2(c)). Finally, the most similar partition is selected to receive the new page p (Figure 2(d)). For a more detailed description of MTD, we refer the interested reader to (VIEIRA et al., 2009).

Figure 2 – Steps performed by MTD for each page p until p is added to a partition whose pages share a common template with p .



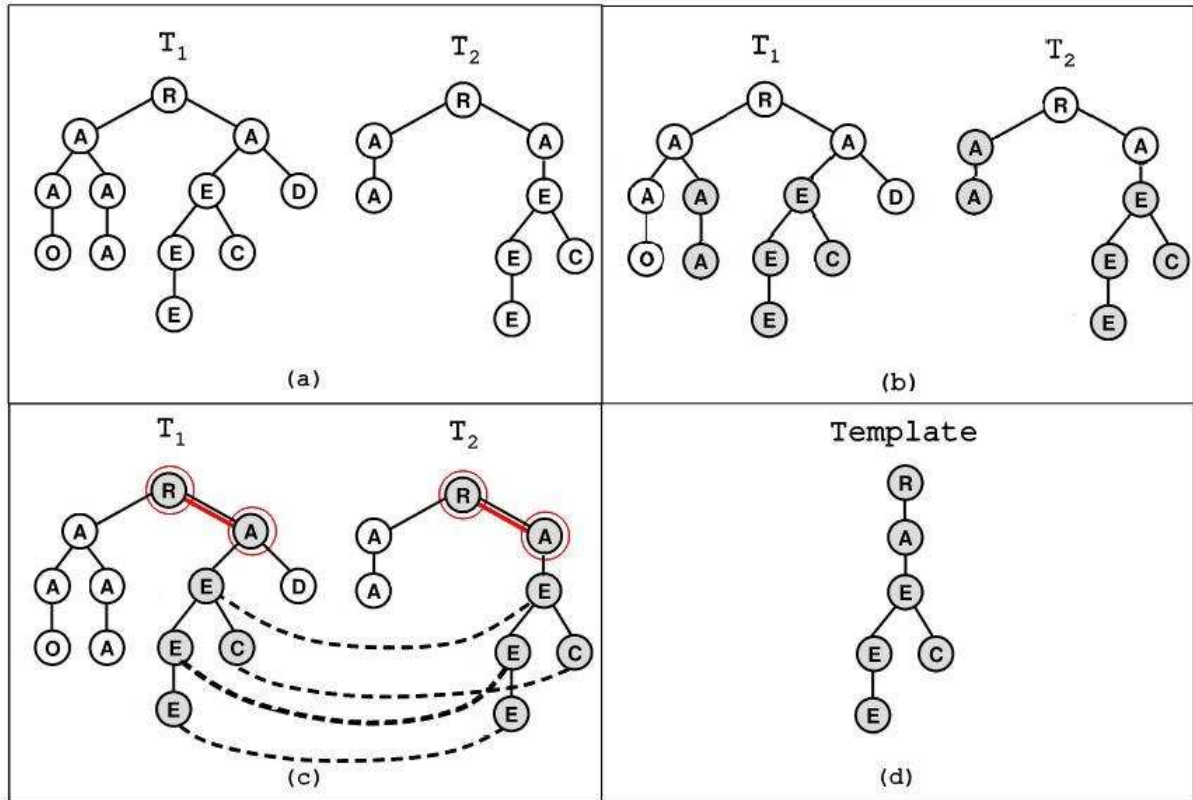
4.2 Algorithm RBM-TD

In this section, we illustrate how algorithm RBM-TD (Restricted Bottom-up Mapping) finds a template through the bottom-up mapping between DOM trees. The complexity of the DOM tree mapping between two trees is $O(|T_1| + |T_2|)$, where $|T_x|$ denotes the number of nodes of tree T_x . This algorithm is derived from the one proposed by Valiente (2001). The main difference to Valiente's algorithm lays in the fact that RBM-TD takes into consideration only subtrees which are co-located in the same tree path starting in the root.

Figure 3 illustrates RBM-TD. Given two DOM trees T_1 and T_2 (Figure 3(a)), the algorithm determines all the sets of identical subtrees which we refer to as equivalence classes (Figure 3(b)). Then, it traverses the trees top-down and match the identical subtrees which have the same path starting in the tree roots, as we can see in Figure 3(c). At the end, the template consists of the set of matched nodes, depicted in grey (Figure 3(d)).

5 EXPERIMENTS

In this section, we present the statistics of the web collections adopted in our experiments. Then, we show and comment the results after application of the method MTD in the proposed scenarios. Finally, we discuss the impact of template removal on search engines.

Figure 3 – Example of bottom up mapping performed by RBM-TD.

5.1 Data Sets

The experiments were carried out in two scenarios: a large scale and intrasite. Next, we show a description of the collections adopted in each scenario. For the experiments involving the removal of templates in large scale scenarios, we adopt the collection GOV2 (CLARKE; CRASWELL; SOBOROFF, 2005). This collection consists of 25,205,179 documents crawled in 2004 from the “.gov” domain. The GOV2 includes both HTML pages and text extracted from PDF, word files and postscripts.

To estimate the search quality in GOV2, we used two set of queries. The first one comprises 150 informational queries available in the GOV2 reference collection and adopted in the 2007 TREC conference (ALLAN et al., 2007); the second one consists of 252 navigational queries adopted in TREC 2004 conference (CLARKE; CRASWELL; SOBOROFF, 2005). Notice that the two set of queries are part of a reference collection, being created with the goal of providing a fair and solid framework for evaluating quality of results provided by Web search systems.

For the intrasite scenario, we carried out the experiments on 4 real Web collections namely IG, CNN, CNET, and BLOGs. These collections were created in a previous work (MOURA et al., 2010) and provide a useful framework to evaluate intrasite search methods. The IG collection, which contains 34,460 pages crawled from one of the largest Brazilian Web

portals (see www.ig.com.br), is composed of a recipe site, a forum site and a news website. The set of test queries used in ranking experiments for this collection is composed of 50 popular queries extracted from a log of queries submitted to the IG Web search service. Relevance assessments for these queries were made by 85 volunteers from 5 different Brazilian universities. A pooling method was adopted by Moura et al. (2010) to collect relevance judgments.

For each of the 50 queries, Moura et al. (2010) created a query pool formed by the union of the top 20 documents retrieved by each ranking method considered in their experiments. To avoid noise and erroneous judgments, each document retrieved by a given query was evaluated by three distinct volunteers. A document was considered as relevant to a query when at least two of the volunteers considered it as relevant to that query. This procedure was also performed in the other three Web collections adopted in the intrasite scenario. However, it is important to note that these evaluations were made only by Moura et al. (2010), and that our experiments retrieved a few non-judged documents in the top 20 answers.

The second collection is a crawling of the CNN Web site composed of 16,257 Web pages. The queries used for experimentation with this collection were proposed by 50 volunteers, such that each volunteer wrote a single query. The third collection was obtained by crawling four Web sites affiliated to the CNET Web portal (see www.cnet.com): CNET News, that provides news, blogs, and special reports about technology; CNET Download, composed by a large set of pages containing free downloads; CNET Shopper, which is a virtual shop of tech products; and CNET Reviews, composed by a set of pages containing reviews of products. This collection contains a total of 352,770 Web pages. The 50 test queries were obtained in similar way as described for the CNN collection.

The fourth collection was obtained by crawling 9 popular blogs from the top popular list presented in Technorati Blog3. This collection contains a total of 54,055 blog posts and 50 test queries, which were generated as in the CNN collection.

5.2 Evaluation Metrics

To evaluate the informational queries, we adopted the metrics P@N, BPREF-10 and MRR. P@N measures the precision at the top N results provided by the system and is usually adopted to evaluate the quality of Web search systems. BPREF-10 is a metric designed to compare information retrieval systems when only a partial set of the query answers is evaluated. This metric was selected because, as described in previous section, our experiments retrieved documents not evaluated by Moura et al. (2010). To evaluate navigational queries we adopted the MRR metric, which measures results where only one correct answer is enough for the user information need. Therefore, it is the most used metric to estimate the quality of search results for navigational queries. Further details about how to compute such metrics can be found in

(BAEZA-YATES; RIBEIRO-NETO, 1999).³

In this section, we detail the experiments conducted with the template detection algorithms in our test collections. In particular, we describe the choices we have made to enhance precision and the results obtained by applying MTD to detect and remove templates.

The MTD efficacy is directly related to its ability to properly build page DOM trees. This task is particularly hard if the algorithm has to deal with malformed HTML code. To avoid such a problem, we first have fixed the pages by applying the HTML fixer CyberNeko4 over all GOV2 pages. After that, we partitioned the GOV2 according its page domains (as in Figure 4), because we believe that pages in a same domain are more likely to have a same template. Note that such a partitioning also contributes to reduce MTD costs since the algorithm has not to be applied to the entire collection of pages at once. After the partitioning, MTD is performed over smaller collections comprising GOV2 domains. As a result of this process, we obtained 1202 groups of pages.

Figure 4 – Example of a collection partitioned into domains.



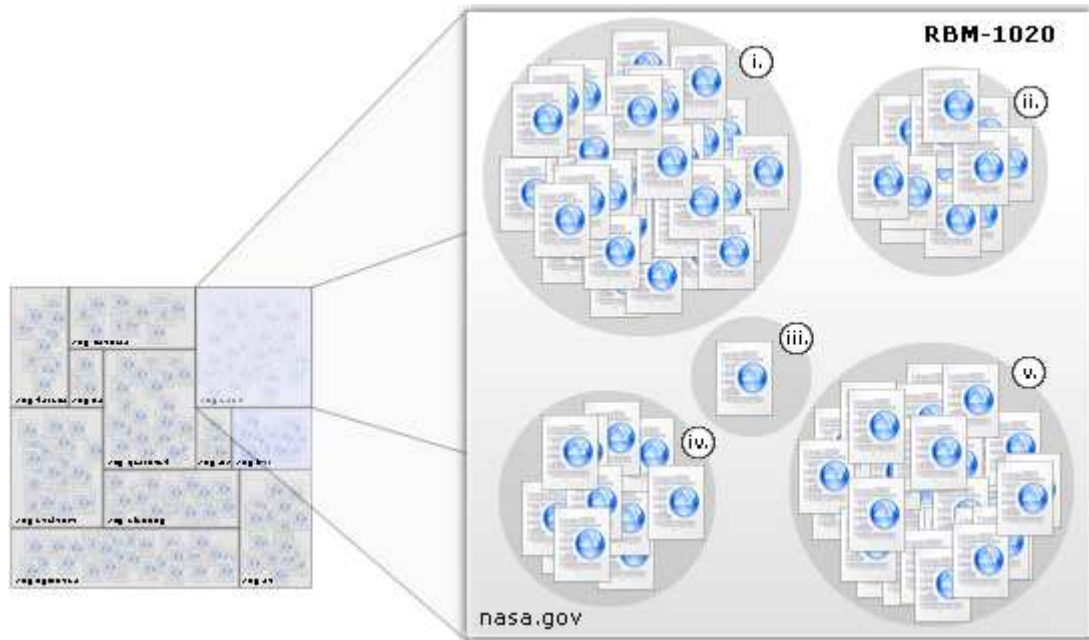
The MTD performance depends on the combination of the parameters *path* and *tree*. These parameters establish a tradeoff between the computational time spent in detection and the quality of the detected templates. Since we have to process a very large number of pages, we first use as *path* and *tree* the values suggested by (VIEIRA et al., 2009) as the most cost-effective in their experiments (i.e. $path = 10$ and $tree = 20$). As previously described, we use RBM-TD for single template detection. From now on, we refer to this specific combination of methods and parameters as RBM-1020. Further, we refer to the index created using pages for which the templates were removed using RBM-1020 as RBM-1020 index.

After partitioning GOV2 into domains, MTD is applied to each partition. This process is illustrated in Figure 5(a), considering the “nasa.gov” domain. The method creates five page groups that probably share the same template. For each group, a template is detected. According to (VIEIRA et al., 2009), RBM-TD is able to correctly identify templates in groups with 10 or

³<http://technorati.com/blogs/top100>

more pages. Thus, in our experiments, we discard templates extracted from groups with less than ten pages. This task is repeated over all partitions until the entire collection has been processed.

Figure 5 – MTD performed in nasa.gov domain.



After running RBM-1020 over all 1202 partitions in GOV2, we identified 67,313 groups with ten or more pages (see Table 1). These groups correspond to about 75% of the pages in the collection. The average size of the groups is about 280 pages. In contrast, when analyzing the groups of less than ten pages (see Table 2), we note that about 21% of the collection is composed of groups with only one page. That is a characteristic of GOV2 which, as a government domain, has many pages which are single text transcripts such as laws, ordinances, acts, etc. Such pages have no templates.

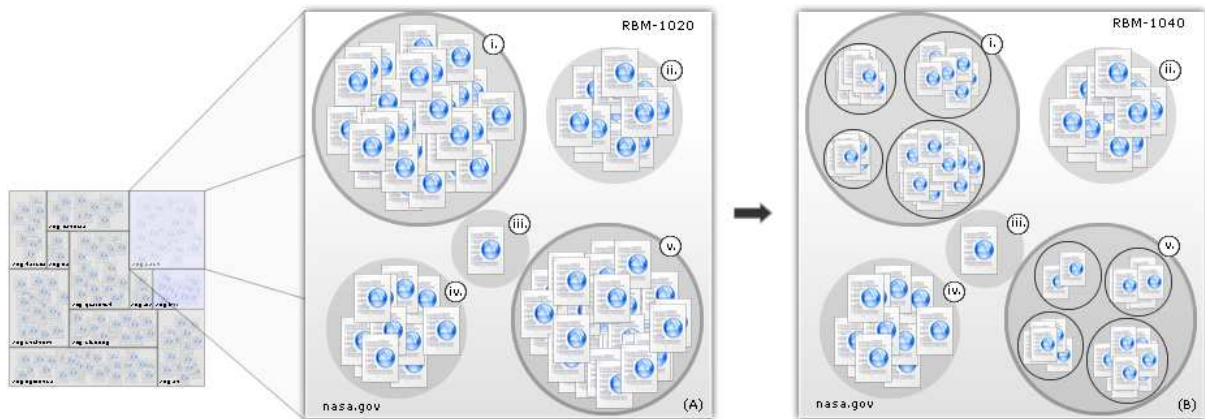
As observed in Table 1, 16.74% of the groups obtained by MTD have empty templates, i.e, templates with no content. That was due to the parameters we used, which led to the formation of groups with a large amount of pages that probably did not share the same template. Such problem may be alleviated by the selection of an appropriate set of parameters aiming at improving the quality of the templates. To accomplish this, we performed MTD with $tree = 40$ over the groups of pages for which MTD previously failed to detect good quality templates. Those groups were found after a manual inspection and consisted of the ones with more than 1000 pages. We refer to this new combination of parameters as RBM-1040. As before, we refer to the index created using pages for which the templates were removed using RBM-1040 as RBM-1040 index.

Figure 4 illustrates the application of RBM-1040 over the “nasa.gov” domain. Consider, in that figure, that only groups I and V (highlighted in Figure 6(a)) have more than 1000 pages after running RBM-1020.

Table 1 – Results obtained after applying MTD to GOV2.

GOV2	RBM-1020	RBM-1040
Groups with 10 or more pages	67,313	76,484
Pages in groups greater or equal to 10	18,874,100	18,509,370
% of the collection in groups of 10 or more pages	74.88	73.43
Average size groups	280.39	242
Empty templates	16.74%	15.194%
Average words in the templates	26.89	27.10

Note in Figure 6(b) that RBM-1040 was performed only on these two groups, which led to smaller groups, i.e, groups with a higher probability of containing pages that share the same template.

Figure 6 – MRBM-1020 and RBM-1040 applied to the nasa.gov domain.**Table 2 – Groups with less than 10 pages**

group size	number of pages
1	5,330,998
2	193,879
3	51,896
4	28,067
5	16,932
6	12,719
7	9,361
8	7,680
9	6,112

As seen in Table 1, the application of RBM-1040 to 1165 groups with more than 1000 pages resulted in the detection of templates of better quality, that is, a larger number of not empty templates with a larger average number of words per template.

5.3 Website Collections

Given the good performance of RBM-1040, we decided to use it also for detecting templates in the Website collections. However, as these collections are small, we did not partition them into domains. Table 3 summarizes the main results obtained after running RBM-1040 over each of these collections.

Table 3 – Results obtained after applying MTD to datasets BLOGs, CNN, CNET, and IG.

Dataset	BLOGs	CNN	CNT	IG
Groups with 10 or more pages	9	9	32	16
Average size groups	6,006	1,806.11	12,164.31	2,153.56
Empty templates	2	2	4	2
Average words in the templates	107.55	182.22	71.34	142.75

6 IMPACT OF TEMPLATE REMOVAL ON WEB SEARCH

In this section, we present an analysis of the impact of template removal in the two scenarios proposed in this paper. The experiments were conducted as follows. First, we detected the templates to be removed using MTD. Next, we removed the templates. Then, we generate two indices for each collection, with and without templates. Finally, we evaluated the impact of the templates by processing queries using the two indices.

The size of the indices built after performing the described procedure is shown in Tables IV and V. From these tables, we note that template removal led to larger index reductions in collections in the intrasite scenario. The best result for the Web scenario was about 10%, and this reduction was worse than the worst result for the intrasite scenario (a reduction of about 14%).

Table 4 – Index size after running MTD in GOV2.

Collections	Index size	Size reduction
Original	177GB	–
RBM-1020	172 GB	2.82%
RBM-1040	160 GB	9.50%

To evaluate the template detection performed by MTD in intrasite scenario, we manually removed the templates from all pages of the IG, CNN, BLOGs and CNET collections. As seen in Table 5, the reductions of the indices of Web sites collections obtained by MTD ranged from 14.20% (CNN) to 34.5% (CNET). When compared to the manual template removal, we note very similar results for Blogs and CNN, and some discrepancy for CNET and IG. More specifically, manual reduction was able to achieve smaller indices for CNET and IG than MTD. Such

result is probably due to the parameters used which were not the best for these two collections. Moreover, the HTML fixer CyberNeko⁴ was not able to eliminate all HTML errors which led to poor template detection.

Table 5 – Index sizes after running MTD in intrasite scenario collections, i.e., BLOGS, CNN, CNET, and IG.

	Collection	Index size (MTD)	Manual
BLOGs	With Template	383 MB	–
	No Template	316 MB	–
	Reduction	17.49%	22.6%
CNN	With Template	169 MB	–
	No Template	145 MB	–
	Reduction	14.20%	12.1%
CNET	With Template	2.53 GB	–
	No Template	2.08 GB	–
	Reduction	17.78%	34.5%
IG	With Template	202 MB	–
	No Template	170 MB	–
	Reduction	15.84%	27.9%

Overall, we note that template removal led to significant index size reductions. Such smaller indices require less space to be stored and are quickly processed since lower inverted lists have to be traversed during query processing. We now evaluate the impact of the new indices on the quality of the search results by submitting queries to the reduced indices. As previously mentioned, some studies in literature have suggested that template removal could improve the search quality since templates would correspond to noisy content.

After processing the set of informational queries in GOV2, we found that the quality of the answers was slightly better when using the original index than when using the reduced one (Table 5). Statistically significant results in that table are indicated by an ‘*’⁵.

As we can see, both parameter sets used, RBM-1020 and RBM-1040, led to significant quality losses. No significant difference in performance was observed when comparing RBM-1040 and RBM-1020.

We now evaluate the impact of the indices without templates when processing navigational queries in GOV2 (see Table 7). For both parameter sets used, we observe a decrease in MRR values. In particular, the number of searched pages among the top ten answers was slightly smaller when using the RBM-1040 index than the RBM-1020 index. Further, 21.4% of the pages searched in the RBM-1020 index were not found whereas 20.6% were not found in the RBM-1040 index.

Table 8 shows the results obtained after removing templates from the collections of websites. Note that, according to the Wilcoxon test, there is no significant difference on using

⁴<http://sourceforge.net/projects/nekhtml>

⁵We test the differences using the Wilcoxon test and a 95% confidence level

Table 6 – Results obtained by informational queries in GOV2, using indices with and without templates. An ‘*’ stands for statistically significant differences according to Wilcoxon test.

Parameters	GOV2	P@5	P@10	Bpref-10
RBM-1020	With Template	0.2940	0.2886	0.2976
	Without Template	0.2859	0.2658	0.2831
	Gain (%)	-0.81	2.28*	1.45*
RBM-1040	With Template	0.2940	0.2886	0.2976
	Without Template	0.2832	0.2705	0.2872
	Gain (%)	-1.08*	-1.81*	-1.04

Table 7 – Results obtained for the navigational set of queries on GOV2 collection.

GOV2	MRR	% Top 10	% Not Found
Original	0.303	44.4 (112)	18.7(47)
RBM-1020	0.295	43.3(109)	21.4(54)
RBM-1040	0.295	41.7 (105)	20.6(52)

or not using templates regarding search quality in collections BLOGs, CNN, and CNET. For IG, we observed significant differences in all metrics. In that case, the removal of templates led to a loss probably due to bad trees derived from malformed HTML code because CyberNeko was not able to fix all the errors.

Table 8 – Results obtained for collections BLOG, CNN, CNET, and IG. An ‘*’ stands for statistically significant differences according to Wilcoxon test.

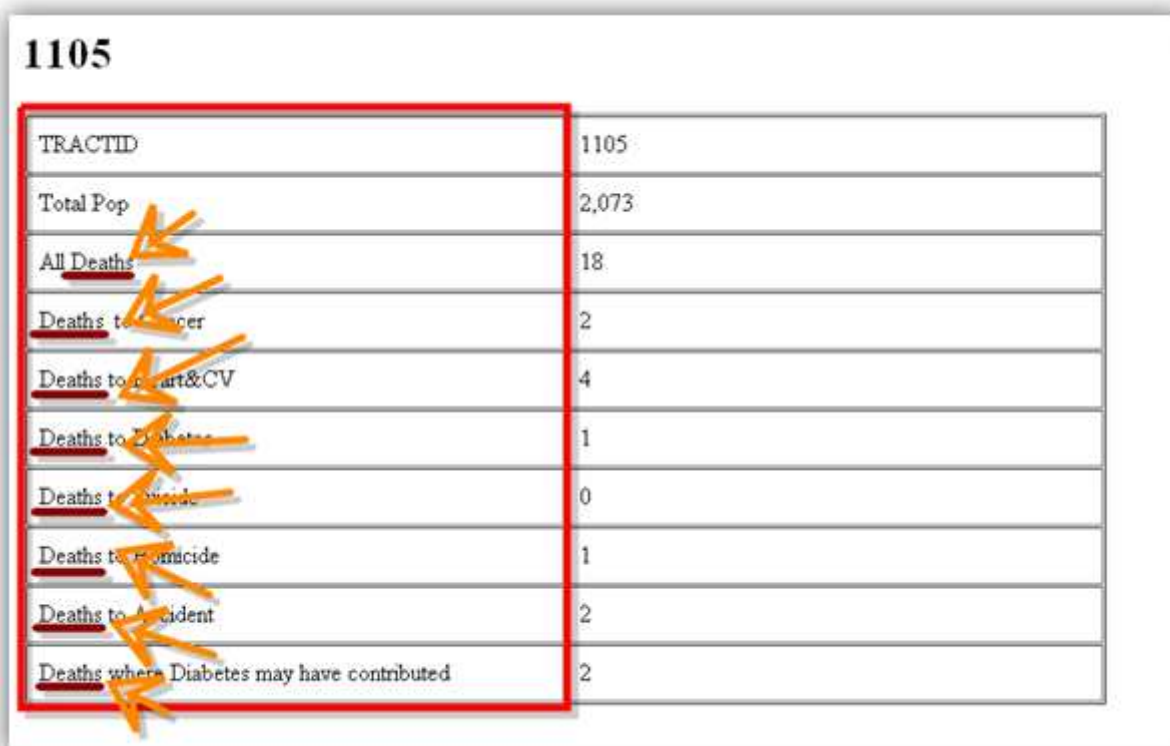
	Collection	P@5	P@10	Bpref-10
BLOGs	With Template	0.6600	0.6180	0.7980
	Without Template	0.6560	0.6000	0.8311
	Gain (%)	-0.4	-1.8	3.31
CNN	With Template	0.7400	0.6520	0.8336
	Without Template	0.7480	0.6600	0.8384
	Gain (%)	0.8	0.8	0.48
CNET	With Template	0.4375	0.4396	0.5957
	Without Template	0.4542	0.4167	0.6454
	Gain (%)	1.67	-2.29	4.97
IG	With Template	0.6840	0.6520	0.7002
	Without Template	0.6320	0.6040	0.6449
	Gain (%)	-5.2*	-4.8*	-5.53*

In spite of no significant difference has been observed for the cases in which the removal of templates led to a larger precision, after a manual inspection, we observed that the performance of some queries is improved. These queries are the ones which contain terms in the template of many pages. Although search engine users are not looking for information in templates, the removal of its terms may hurt the recovery of some pages considered relevant to the query. By removing the template, we change one of the main factors of the similarity

equation used to match a page and a query, the term frequency (TF)⁶. To better understand the situations involving template removal on search engines, we illustrate some cases identified in our tested scenarios.

Figure 7 shows a page retrieved for the GOV2 informational query “hunting-deaths”. The template of the page corresponds to the red-framed box and the query terms are highlighted by arrows. The depicted page and all the other pages which share that same template were considered non relevant to the query. After a careful inspection, we note that one of the query terms present in the template has a high TF value, which led to that page being placed among the top answers, hurting the search quality. In such a case, the template removal would be advantageous.

Figure 7 – Example of non relevant page for the query “hunting deaths”.



TRACTID	1105
Total Pop	2,073
All Deaths	18
Deaths to Cancer	2
Deaths to Heart&CV	4
Deaths to Diabetes	1
Deaths to Suicide	0
Deaths to Homicide	1
Deaths to Accident	2
Deaths where Diabetes may have contributed	2

We now present a case of common template to a large number of pages in collection CNET (Figure 8). Among other features, this template includes a list of popular site topics. Because of that list, many non relevant pages are retrieved as answers to the query “ipod nano”. As before, in such a situation, removing templates would be advantageous.

Unlike the previous two examples, the next ones highlight situations where template removal would be harmful. Figure 9 shows an answer to query “1890 census” in GOV2 index

⁶Note that, besides the TF, another important factor for computing the similarity between query and page is the page norm, that is, the quantity of information found in a page. By removing the templates, we modify the page norm and, by extent, we reduce the similarity of a page to a query. Thus, the set of answers provided by the search system may change as a consequence of a template removal, even if the query terms are not present in the template.

Figure 8 – Example of noisy template shared by many pages in CNET.

(templates not removed). All pages that shared that template were considered relevant to the query, because they present information about the census of the city in 1890. Thus, the removal of that template would lead to miss useful pages.

Finally, Figure 10 shows a page retrieved for query “big dig pork” in GOV2. That page and a large number of pages that share the highlighted template are considered relevant to the query. As we note in Figure 10, the all the query terms in that page are found in the template. Thus, its removal would certainly lead to the search system misses that page.

7 CONCLUSION

In this paper, we present a comprehensive study about the impact of template removal techniques to rank quality in search systems. Several previous works indicate that templates may act as noisy information to search systems and, thus, their removal could improve the overall quality of such systems. However, there was a lack of efforts in literature to better assert such impact.

Our experiments suggest that the simple removal of templates from Web pages does not necessarily lead to improvements in ranking quality. We investigated the application of template removal in two distinct scenarios and in both cases we could not observe any gain in ranking quality. As we show in our experiments, the most effective impact of template removal to search systems in these two scenarios is the reduction in the amount of information indexed, which shows that template removal could be used as an interesting loss compression technique. On the other hand, current methods to detect and remove templates present high computational costs and their application as a loss compression method probably not worth the gains.

Regardless of the negative conclusions about the impact of template removal to the overall search quality, a detailed analysis of the cases suggests that its application can be worthwhile for some types of queries. In these cases, template information should be used as an auxiliary source of relevance information, instead of just being removed. We plan to investigate this alternative as future work.

Figure 9 – Example of page relevant to the query “1890 census”.

Caroline County

Click on the pad to see a larger version
[MD map showing location of Caroline County]

Caroline County was created in 1773 from Dorchester and Queen Anne's counties, and was named for Lady Caroline Eden. She was the wife of Maryland's last colonial governor, Robert Eden (1741-1784); the daughter of Charles Calvert, fifth Lord Baltimore; and the sister of Frederick Calvert, sixth Lord Baltimore.

The Latin motto, *Terra Dulcis Vivendum* means *Land of Pleasant Living*. The county seat is Denton.

- County Map
 - [1895](#)
 - [Today](#)
- [Courthouses](#)
- Histories
 - [Caroline County](#)
 - [Denton](#)
 - [Preston](#)
 - [Kent Isle](#), Maryland's First Settlement
 - [Postcards](#), a wonderful collection from the Caroline Co. Library
- [Tire Playground](#)
- Major Industries - Agriculture
- Attractions - [Adkins Arboretum](#), [Tuckahoe State Park](#)
- [Places to Visit in Caroline County](#)

Below is some interesting census information about Caroline County. Look at the way the county population has changed over the years!

Population Information for Caroline County		
	Population	% of MD Population
1790 Census	9,506	3.0%
1890 Census	13,903	1.3%
1990 Census	27,035	.6%
2000 Census	29,772	.6%

© Copyright October 16, 1997, Office of the Secretary of State. Last Modified 3/01

Figure 10 – Example of page relevant to the query “big dig pork”.

Central Artery/Third Harbor Tunnel Project
Oversight Coordination Commission

Commission Summary Report July 1998

[Contents](#)

[Background](#)

[The Oversight Coordination Commission](#)

[Summary of Work](#)

[Conclusion](#)

CONCLUSION

As the Commission begins its second year of operation, continued emphasis will be placed on communicating ideas, sharing information, and coordinating the oversight activities of the OAG, OSA, and OIG. The Commission will continue to provide meaningful and timely oversight to help ensure that the CAT Project is adequately protected from waste, fraud, and abuse. The Commission remains committed to aggressive, proactive, independent state oversight of the CAT Project.

Additional copies of this Summary Report are available and can be obtained by a request in writing to:

Central Artery/Third Harbor Tunnel Oversight Coordination Commission
State House Station
P.O. Box 250
Boston, Massachusetts 02133

[Commission] [Attorney General] [State Auditor] [Inspector General] [\[Big Dig Fraud Hotline\]](#)

REFERENCES

- ALLAN, J. et al. Million query track 2007 overview. In: 6TH TEXT RETRIEVAL CONFERENCE (TREC 2007 - NIST ST500 - 274). **Proceedings of the NIST**. Gaithersburg, Maryland, USA, 2007.
- BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., 1999.
- BARYOSSEF, Z.; RAJAGOPALAN, S. Template detection via data mining and its applications. In: ELEVENTH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 2002. **Proceedings of the...** New York, NY, USA: ACM, 2002. p. 580 – 591.
- CAI, D. et al. **Vips: a vision-based page segmentation algorithm**. MICROSOFT TECHNICAL REPORT. 2003.
- CHEN, L.; YE, S.; LI, X. Template detection for large scale search engines. In: TWENTY FIRST ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING (SAC'06), 2006. **Proceedings of the...** New York, NY, USA: ACM, 2006. p. 1094–1098.
- CLARKE, Charles; CRASWELL, Nick; SOBOROFF, Ian. The trec terabyte retrieval track. **SIGIR Forum**, ACM, New York, NY, USA, v. 39, n. 1, p. 25–25, jun 2005. ISSN 0163-5840. Disponível em: <<http://doi.acm.org/10.1145/1067268.1067274>>.
- FERNANDES, D. et al. Computing block importance for searching on web sites. In: SIXTEENTH ACM CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT. CIKM'07, 2007. **Proceedings of the...** New York, NY, USA: ACM, 2007. p. 165–174.
- GIBSON, D.; PUNERA, K.; TOMKINS, A. The volume and evolution of web page templates. In: FOURTEENTH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB. WWW'05, 2005. **Special interest tracks and posters of the**. New York, NY, USA: ACM, 2005. p. 830–839.
- MA, L. et al. Extracting unstructured data from template generated web documents. In: TWELFTH INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, CIKM'03, 2002. **Proceedings of the...** New York, NY, USA: ACM, 2003. p. 512 – 515.
- MOURA, E. S. et al. Using structural information to improve search in web collections. **J. Am. Soc. Inf. Sci. Technol**, v. 61, p. 2503–2513, dec. 2010.
- SONG, R. et al. Learning block importance models for web pages. In: THIRD INTERNATIONAL CONFERENCE ON WORLD WIDE WEB. WWW'04, 2004. **Proceedings of the...** New York, NY, USA: ACM, 2004. p. 203–211.
- VALIENTE, G. An efficient bottom-up distance between trees. In: EIGHTH INTERNATIONAL SYMPOSIUM OF STRING PROCESSING AND INFORMATION RETRIEVAL, 2001. **Proceedings of the...** Washington: IEEE Computer Science Press, 2001. p. 212–219.
- VIEIRA, Karane et al. On finding templates on web collections. **World Wide Web**, Kluwer Academic Publishers, Hingham, MA, USA, v. 12, n. 2, p. 171–211, jun. 2009. ISSN 1386-145X. Disponível em: <<http://dx.doi.org/10.1007/s11280-009-0059-3>>.

VIEIRA, K.; PINTO, A. S. A fast and robust method for web page template detection and removal. In: FIFTEENTH ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 2006. **Proceedings of the CIKM**. New York, NY, USA: ACM, 2006. p. 258–267.

VIEIRA, K.; SILVA, A. S. **Detecção e extração de templates em páginas web**. 2008. Dissertação (Mestrado) — Universidade Federal do Amazonas, Programa de Pós-graduação em Informática.

WAN, S.; THOMAS, P.; ROWLANDS, T. Web indexing on a diet: Template removal with the sandwich algorithm. In: FOURTEENTH AUSTRALASIAN DOCUMENT COMPUTING SYMPOSIUM, 2009. **Proceedings of the...** New York, NY, USA: ACM, 2009. p. 296–305.

YI, L.; LIU, B.; LI, X. Eliminating noisy information in web pages for data mining. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2003. **Proceedings of the ninth ACM SIGKDD**. New York, NY, USA: ACM, 2003. p. 296–305.