# A MULTI-SEQUENCE ALIGNMENT ALGORITHM FOR WEB TEMPLATE DETECTION

Filippo Geraci[1] and Marco Maggini[2]
[1]*Istituto di Informatica e Telematica, CNR, Pisa, Italy*
[2]*Dipartimento di Ingegneria dell'Informazione, Universitá di Siena, Siena, Italy*
*filippo.geraci@iit.cnr.it, maggini@ing.unisi.it*

Keywords: Web information retrieval, Web template detection.

Abstract: Nowadays most of Web pages are automatically assembled by content management systems or editing tools that apply a fixed template to give a uniform structure to all the documents beloging to the same site. The template usually contains side information that provides better graphics, navigation bars and menus, banners and advertisements that are aimed to improve the users' browsing experience but may hinder tools for automatic processing of Web documents. In this paper, we present a novel template removing technique that exploits a sequence alignment algorithm from bioinformatics that is able to automatically extract the template from a quite small sample of pages from the same site. The algorithm detects the common structure of HTML tags among pairs of pages and merges the partial hypotheses using a binary tree consensus schema. The experimental results show that the algorithm is able to attain a good precision and recall in the retrieval of the real template structure exploiting just 16 sample pages from the site. Moreover, the positive impact of the template removing technique is shown on a Web page clustering task.

## 1 INTRODUCTION

Automatic processing of Web contents is nowadays an important tool to provide added value services to users. For instance, search engines have become an indispensable and powerful gateway to access information on the Web, information extraction techniques are the basis to design systems for data integration and collection from unstructured Web documents, sentiment analysis and topic monitoring from Web posts are attracting an increasing interest from companies. Most of these applications need to focus only on the most informative part of a Web document and the quality of their results may be heavily hindered by the large amount of side information that is usually embedded in Web documents. In fact, most of Web pages are usually dynamically constructed by content management systems or edited with design tools that use a fixed template to format them. Basically a template provides a common structure of all the pages that is reflected both in the graphical layout and in the skeleton of HTML tags that are used to format the page.

Many different techniques for template detection and removal have been proposed in the literature. All the reported results show that automatic processing techniques, like indexing, Web page classification and clustering, may benefit from template removal. Some early works approached the task by a layout analysis based on a graphical rendering of the page but their practical use is limited by the high computational costs (Kovacevic et al., 2002). The most recent algorithms try to exploit the structure of HTML tags by searching for common parts of the pages or blocks (Bar-Yossef and Rajagopalan, 2002; Debnath et al., 2005; Song et al., 2004) or by an analysis of the document tree as defined by HTML tags (Yi et al., 2003; Chakrabarti et al., 2007). A related application that exploits similar techniques is wrapper induction (see e.g. (Wong and Lam, 2007)), that aims at the automatic generation of information extraction modules by exploiting the common structure of the Web pages containing the data we are interested to collect.

Desirable features of an automatic template extraction algorithms are low computational costs since they may employed in large scale systems, easy adaptability to any template structure using a minimum number of sample pages, the use of a completely unsupervised (or a very limited supervised) approach to avoid the need of human effort that would limit its applicability, and high precision in the detection of the template structure. In this paper we present a template extraction algorithm that meets most of these requirements. The proposed technique exploits the Needleman and Wunsch alignment algorithm originally devised for DNA sequences (Needlemana and Wunscha, 1970). This algorithm is exploited to find the most likely alignment among two sequences of HTML tags extracted from two different pages from the same site. Since the common sequence structure that is extracted from a pair of pages may be strongly dependent on the specific selected pages, a hierarchical consensus schema is adopted: given a certain number of different pairs of pages, the extracted common sequences, that constitute the current set of hypotheses for the target template, are recursively compared to further distillate the common parts. The comparisons form a binary consensus tree and the output of the algorithm is the template sequence available in the tree root. The experimental results show that the proposed method is able to yield a good precision and recall in the detection of the HTML tags belonging to the template given only a very limited number of sample pages from the site (in the considered setting only 16 pages are able to provide satisfactory performances). The algorithm is also efficient and the running time for template extraction is quite low. Finally, the evaluation shows that the proposed extraction technique can provide significant benefits in a Web mining task, namely Web page clustering, confirming similar results as those reported in the literature.

The paper is organized as follows. The next section describes the template extraction algorithm in details. Then section 3 reports both the evaluation of the accuracy in the prediction of the real template and an analysis of the impact of the template removal in a Web mining application (Web page clustering). Finally in section 4 the conclusions are drawn and the future developments are sketched.

## 2 TEMPLATE DETECTION ALGORITHM

The proposed template extraction algorithm is based on the assumption that a Web template is made up of a set of overrepresented contiguous subsequences of HTML tags shared by the pages from a given web site. According to this assumption, some local parts of the template can be missing in some particular web page.

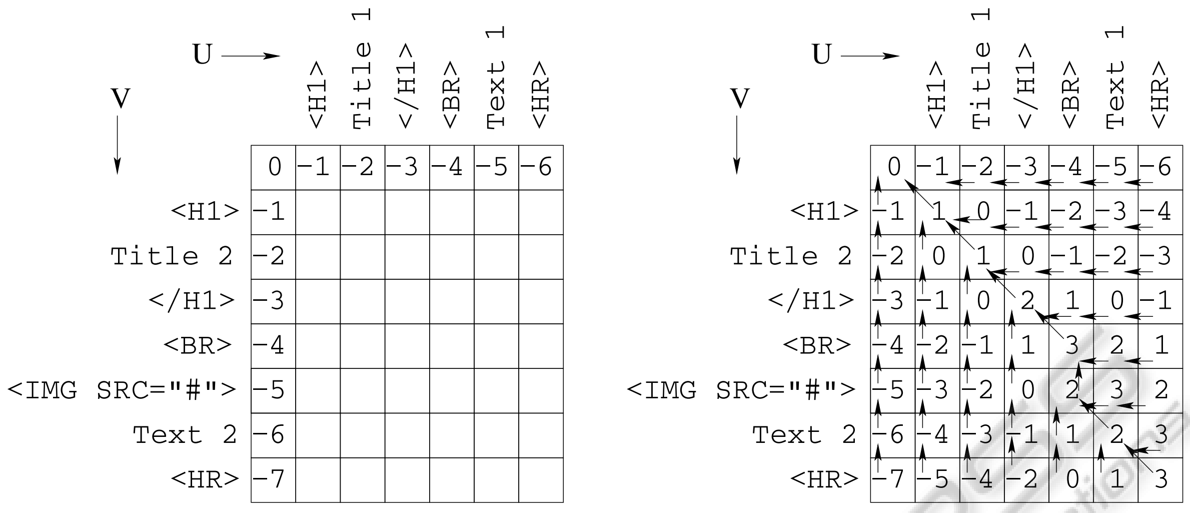
The first processing step consists in splitting each page into a sequence of tokens. In the following, a token corresponds to a single HTML tag or to the text snippet contained between two consecutive tags. The tokens are then normalized to remove the potential differences due to the human editing of the pages or irrelevant features (i.e. extra white spaces are removed, capital letters are lowered). Moreover tags are normalized by sorting their internal attributes. The normalized tokens form an alphabet of symbols over which the page sequences are generated. Hence, the problem of finding the common parts in two Web pages can be cast as the computation of the global alignment of the two corresponding sequences of normalized tokens. The alignment can be obtained by exploiting a modified version of the DNA global alignment algorithm due to Needleman and Wunsch (Needlemana and Wunscha, 1970), which is based on a dynamic programming technique.

However, a single alignment of only two Web pages is likely to produce a poor approximation of the template since they can share the same tokens just by chance or a portion of the real template could be missing in one of the two pages. Hence, in order to compute a more reliable template, the alignment procedure is repeated exploiting a set $t$ pages using a recursive procedure. At each step the $k$ input sequences are paired and aligned to yield $k/2$ template profile candidates, starting from the initial $t$ sequences representing the available samples. The template candidate originating from a given pair of sequences is obtained by pruning those tokens that have receive low evidence in the alignment steps performed so far. The procedure iterates until it remains only a single profile that is returned as final Web template (i.e. $\log_2(t)$ steps are required).

### 2.1 The Alignment Algorithm

Two strings $s_1$ and $s_2$ are aligned by inserting white spaces (gaps) into them, such that the probability of finding the same symbol in the same position of the two modified strings is maximized, and that only one string can contain a gap in a certain position. An alignment corresponds to the sequence of operations required to align the two strings. Consider the case in which we are comparing the $i$-th symbol of string $s_1$ and the $j$-th symbol of string $s_2$ (later referred to as $s_1[i]$ and $s_2[j]$). There are three options: skip both symbols and move to the next one for both the strings; insert a gap in $s_1[i]$ or in $s_2[j]$. We assign a score (or a penalty) to each operation so as to compute a global score for the whole alignment. In particular we reward the case in which $s_1[i] = s_2[j]$ giving score 1, we ignore the case in which we skip $s_1[i]$ and $s_2[j]$ because they are different, and we penalize the insertion of a gap giving score -1. The goal is to find an alignment with highest score among all the possible alignments.

In the considered application, the strings cor-



(a) Matrix initialization (b) Filled matrix

Figure 1: Similarity matrix for two small HTML snippets.

respond to sequences of tokens extracted from the HTML pages, as described before. Let $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_m\}$ be the sequences of tokens representing two Web pages. The alignment algorithm builds a $(n+1) \times (m+1)$ matrix $S$ (called *similarity matrix*) such that each entry contains the score of the best alignment of a prefix of $U$ and $V$. We initialize the first row and column of $S$ as follows: $S[0][i] = S[i][0] = -i$. Figure 1(a) shows the matrix $S$ initialized for a small portion of two HTML pages.

Once initializated, we fill each line of the similarity matrix $S$ from left to right. Consider the case in which we are filling the cell in position $i, j$. If we insert a gap in position $i$ of $U$, the partial score of the alignment becomes $S[j][i-1] - 1$. Symmetrically, if we insert a gap in position $j$ of the sequence $V$, the partial alignment score becomes $S[j-1][i] - 1$. If we align $v_j$ and $u_i$ there are two cases: the two tokens match, thus score becomes $S[j-1][i-1] + 1$, the tokens do not match, thus the score does not change. Since we are interested in the best possible alignment, the algorithm chooses the alternative that maximizes the score. Note that the value of $S[i][j]$ is always derived from one of its neighbours. In a separate matrix $\hat{S}$ we keep trace of the path from a cell of $S$ to its predecessor. Figure 1(b) shows the final similarity matrix and a visual representation of the corresponding $\hat{S}$.

The final phase of the algorithm navigates $\hat{S}$ from the bottom right position to the top left so as to extract the aligned tokens. Given the element $\hat{S}[j][i]$, if its predecessor is $\hat{S}[j][i-1]$ the algorithm outputs $v_i$, symmetrically, if the predecessor is $\hat{S}[j-1][i]$, the algorithm outputs $u_j$. When the predecessor of $\hat{S}[j][i]$ is $\hat{S}[j-1][i-1]$ we have a match/missmatch. In the first case we output indifferently $u_j$ or $v_j$, otherwise we output both tokens. Note that this sequence of tokens has to be inverted to match the original order.

## 2.2 The Template Distillation Algorithm

Given a set of $t$ input pages represented by their normalized token sequences $s_i^0$, $i = 1, \dots, t$, we can randomly pair these pages to obtain $t/2$ pairs $(s_i^0, s_j^0)$. Each token in these sequences is assigned a score value that accounts the frequency of the token at each given position in the sample pages. Hence, at the first step the tokens are all assigned the initial value of $1/t$, since they are observed in just one case. The output of the alignment algorithm applied on each of these pairs is an aligned sequence $s_k^1, k = 1, \dots, t/2$ containing the union of the two original sequences. The confidence score of each token in the output sequence is computed as the sum of the scores available in the input sequences for the matching tokens, whereas if a token does not match with a token in the other sequence its original score is left unchanged.

The procedure is repeated at each step $r$ using as input the sequences $s_k^{r-1}, k = 1, \dots, t/2^{r-1}$, computed after the alignment and a token pruning phase in the previous step. The pruning phase aims at removing the less frequent tokens from the sequences and, in particular, all the tokens having a score below the value $2^{r-2}/t$ are pruned (given the initial values of the scores, for $r = 1, 2$ no tokens are actually removed). Without loss of generality, we can assume that $t = 2^n$ such that the algorithm terminates after $n = \log_2(t)$ steps when the final sequence corresponding to the output template is computed. The distilla-

Table 1: The Web sites collected in the benchmark dataset, with the related average number of tokens in a typical page and in the corresponding template.

| Web site | # tokens | # template tokens |
|---|---|---|
| `PI: punto-informatico.it` | 876 | 686 |
| `Ansa: ansa.it` | 2245 | 1794 |
| `Wikipedia: en.wikipedia.org` | 2058 | 467 |
| `allMusic: allmusic.com` | 1559 | 282 |
| `zdNet: zdnet.com` | 2882 | 2820 |

tion procedure can be described as a binary tree whose leaves correspond to the input pages whereas the internal nodes are the result of the recursive distillation process. Nodes in the higher levels represent more reliable template hypotheses since they collect the evidence extracted by combining the alignments originating from a larger set of pages. Finally the final result is the represented by the sequence available at the root of the tree. Hence, it is easy to verify that the procedure requires to perform a total of $2^n - 1 = t - 1$ alignments.

The sequence available at the root node is finally pruned by removing those tokens whose score is below $1/2$, that is that the evidence of their presence is supported by less than half of the pages used in the process (this is the value used in the experiments but a different threshold could be used).

## 3 EXPERIMENTAL RESULTS

The computational cost and the impact of the template extraction algorithm in a typical Web mining task were evaluated in a set of experiments. In particular, the first evaluation was designed to assess the ability of the algorithm to predict exactly which tokens belong to the template for a given Web site, whereas a second benchmark has been performed to investigate the contribution of the algorithm on the final outcome of an information retrieval task (namely Web page clustering).

The experiments were performed on a Mac equipped with a 3.33 Ghz Intel core 2 Duo, 8Gb of RAM and the Mac OS X v. 10.6.7 operating system. The algorithm was implemented in Python. Even if we could speed up the algorithm by exploiting the intrinsic parallelism of the CPU by performing many alignments of pages in parallel, we decided to implement the algorithm as a sequential program to avoid biases in the running time evaluation due to the particular CPU architecture. However we would like to remark that, using an architecture with $t/2$ processors, it is possible to speed up the algorithm from $O(t)$ to $O(\log_2(t))$.

### 3.1 Evaluation of Template Prediction

The main goal of the first set of experiments was to assess the ability of the algorithm to predict which tokens belong to the Web site template. To the best of our knowledge there is not a public available dataset of Web templates to be used for the comparison of template extraction algorithms. To perform an accurate evaluation the dataset should contain a set of pages extracted from Web sites that exploit different template structures and for each of them the actual template, that is the ground truth, should be available. Hence, we decided to select five Web sites with different size and characteristics in their templates (i.e. the ratio between the number of template tokens and the overall number of tokens in a typical page). For each of them, the "true" template was manually extracted. The sites we selected are reported in table 1, that shows for each of them the average number of tokens per page and the number of tokens in the associated template.

The task of extracting the template from a Web page (or a set of pages) is made complex because of two reasons:

- to discriminate whether a certain section of HTML is part or not of the template can result in an arbitrary choice;
- the template can contain some tags which require to be manually modified.

The first situation can be found in Web sites that are divided into sections, where each section shares with the others most of the template except some details. In these cases, we could define a more general template for the whole site or more specific templates for each section. Another interesting case is that of on–line newspapers in which often the home page has a slightly different structure than the other pages. An example of the second case is a template featuring a context menu such that each page contains a link pointing to itself or to the category to which it belongs. Another case is when the pages contain random generated banners/ads for which the template allocates some space, but the target link changes every time.

Table 2: Precision, recall and number of tags in the extracted template for different values of the parameter $t$.

| | `PI: punto-informatico.it` | | | | | `Ansa: ansa.it` | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| # Pages | 2 | 4 | 8 | 16 | 32 | 2 | 4 | 8 | 16 | 32 |
| Precision | 0.720 | 0.804 | 0.842 | 0.855 | 0.811 | 0.681 | 0.816 | 0.891 | 0.856 | 0.859 |
| Recall | 0.940 | 0.940 | 0.930 | 0.868 | 0.739 | 0.973 | 0.957 | 0.903 | 0.908 | 0.892 |
| # tags | 915 | 808 | 757 | 697 | 625 | 2651 | 2159 | 1822 | 1909 | 1869 |
| | `Wikipedia: en.wikipedia.org` | | | | | `allMusic: allmusic.com` | | | | |
| # Pages | 2 | 4 | 8 | 16 | 32 | 2 | 4 | 8 | 16 | 32 |
| Precision | 0.155 | 0.226 | 0.569 | 0.906 | 0.911 | 0.119 | 0.196 | 0.395 | 0.702 | 0.725 |
| Recall | 0.946 | 0.905 | 0.899 | 0.931 | 0.907 | 0.950 | 0.946 | 0.939 | 0.936 | 0.936 |
| # tags | 3012 | 1864 | 738 | 480 | 466 | 2242 | 1362 | 671 | 377 | 365 |
| | `zdNet: zdnet.com` | | | | | | | | | |
| # Pages | 2 | 4 | 8 | 16 | 32 | | | | | |
| Precision | 0.674 | 0.874 | 0.890 | 0.934 | 0.959 | | | | | |
| Recall | 0.941 | 0.917 | 0.915 | 0.910 | 0.777 | | | | | |
| # tags | 4051 | 2961 | 2908 | 2748 | 2285 | | | | | |

Table 2 reports the precision and recall of the computed template against the manually extracted ground truth, for different values of the parameter $t$. Even if there is not a theoretical limitation for the value of parameter $t$ we narrowed the tests only to powers of 2.

It is not surprising that the recall tends to have a monotonically decreasing trend with respect to the value of the parameter $t$. In fact, when using more pages it is likely to evidence slight differences in the use of the template tags that prevent the algorithm to select them. In contrast the precision tends to increase. This is due to the fact that the higher is the number of pages exploited in the computation of the consensus, the lower is the probability for a token not belonging to the template to be shared enough to be kept in the final hypothesis. On the other hand, negative effects in the precision due to the local discrepancy of some pages are also mitigated by the higher number of pages.

With the purpose of better investigating the effects of the choice of $t$ in the final template prediction quality, we report in figure 2(a) the f-measure for the experiments described above. F-measure is the harmonic mean of precision and recall, and it is commonly used in information retrieval to evaluate the trade-off between these two measures. The plot shows how a overestimated value for $t$ tends to slightly reduce the overall F-measure (as highlighted from `PI` and `zdnet`). On the other hand, a underestimated value of $t$ can result in a poor F-measure (as highlighted from `wikipedia` and `allmusic`) due to a low precision.

We observed that precision is strongly affected by the ratio between the size of the template and the number of tokens in the considered web pages. In particular, for a given assignment of $t$, the higher the ratio, the higher the precision. We observed also that, for increasing values of $t$, the number of tokens returned from the algorithm decreases (see table 2). This latter observation suggests that precision is not influenced by the number of retrieved tokens effectively belonging to the template, but from the number of false positives.

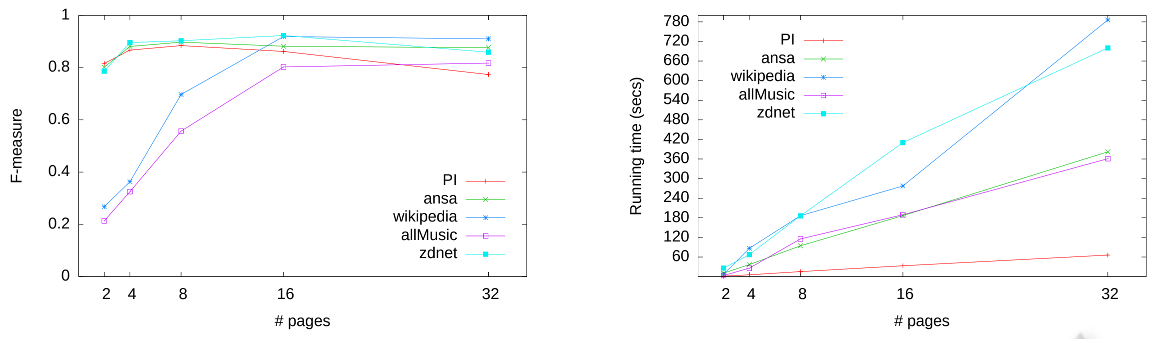
According to Figure 2(a) we can conclude that setting $t = 16$ seems to be a good choice in most situations, at least when an estimation of the web template size can not be automatically accomplished.

Figure 2(b) reports the running time of the algorithm using the hardware and software configuration described earlier in this section. Not surprisingly the algorithm running time depends linearly on $t$. Note that for the assignment of $t = 16$ the algorithm is always able to extract the template in few minutes.

We expected also to find a strong correlation between the running time and the characteristics of the Web sites (average number of tokens, template size), but our experiments confirm only partially this correlation. Consider the cases of `allMusic` and `Ansa`, the former has a lower number of tokens per page and the template size is smaller, however the running time is comparable to the latter. The explanation of this fact is that the running time is affected by the number of tokens in each alignment and not by either the page size or the template size. This feature depends from intrinsic characteristics of the Web pages, thus it is unpredictable in advance.

## 3.2 Web Page Clustering

The main application fields of Web template extraction algorithms are: information retrieval and Web mining. Experimental results in the literature have shown how these algorithms can improve the perfor-



(a) F-measure

(b) Running time

Figure 2: Template extraction algorithm performances with respect to the parameter $t$.

mance in many applications like clustering, classification and indexing (Yi et al., 2003; Bar-Yossef and Rajagopalan, 2002). In this context, the exact identification of the Web template itself is not the main goal, but it is crucial to remove from the document corpus those repeated parts which are not discriminative for the specific task, even if they are not properly part of the Web template.

In this section we report experiments aimed at evaluating the effects of the template extraction algorithm in a clustering task. To this purpose we set up two datasets to which we applied the clustering procedure before and after the removal of the template. We also compare our algorithm with the naive baseline algorithm which removes the real (manually extracted) template.

In table 3 we report some details about the two document corpora used. Both these corpora were obtained downloading a certain number of articles from an Italian on-line magazine. We only downloaded articles and not other kinds of pages (i.e. the home page). This choice has two advantages: first it reduces the potential effects due to the use of slightly different templates in different categories of pages; second, the subdivision of the articles into categories can be easily derived from the manual classification made by the web site owners.

Table 3: Datasets used in the Web page clustering evaluation.

| Dataset | PI | Ansa |
|---|---|---|
| Web site | punto-informatico.it | ansa.it |
| # articles | 9364 | 3470 |
| # Categories | 7 | 13 |

As clustering procedure we used an implementation of the *Furthest-Point-First* (FPF) algorithm (Gonzalez, 1985), which has already been used for related Web mining tasks (Geraci et al., 2008). As quality measures we used four standard functions commonly used in IR to evaluate clustering quality: F-measure, purity, normalized mutual information (NMI), and normalized complementary entropy (NCE).

Let $GT = \{GT_1, \dots, GT_k\}$ be the partition of the articles as extracted by the original classification in the Web sites, and let $C = \{c_1, \dots, c_k\}$ be the outcome of the clustering procedure. Given a cluster $c_j$ and a class $GT_i$, precision and recall can be defined as

$$p_{i,j} = \frac{|GT_i \cap c_j|}{|c_j|}, \qquad r_{i,j} = \frac{|GT_i \cap c_j|}{|GT_i|} \ .$$

Let $F(GT_i, c_j)$ be the F-measure for a given cluster $c_j$ and a class $GT_i$ defined as the harmonic mean of precision and recall. We can extend the concept of F-measure to the entire clustering by using the following formula:

$$F = \sum_i \frac{|GT_i|}{n} \max_j (F(GT_i, c_j)),$$

where $n$ is the sum of the cardinality of all the classes. The value of $F$ is in the range $[0,1]$ and a higher value indicates better quality.

Purity is devised to evaluate how well each cluster fits in a class of the ground truth. Each cluster is assigned to the class which share the highest number of articles with it. The higher number of shared documents the higher cluster quality according to purity. The overall purity for a clustering is the sum of the contributions provided by each cluster. In details purity is defined as:

$$A = \sum_j \frac{|c_j|}{n} \max_i |GT_i \cap c_j| \ .$$

The value of $A$ is in the range $[0,1]$ and a higher value indicates better quality.

The normalized mutual information (see e.g. (Strehl, 2002, page 110)), comes from information theory and is defined as follows:

$$NMI(C,GT) = \\ = \frac{2}{\log|C||GT|}\sum_{c\in C}\sum_{c'\in GT} P(c,c')\cdot\log\frac{P(c,c')}{P(c)\cdot P(c')}$$

where $P(c)$ represents the probability that a randomly selected article $o_j$ belongs to $c$, and $P(c,c')$ represents the probability that a randomly selected article $o_j$ belongs to both $c$ and $c'$. The normalization, achieved by the $\frac{2}{\log|C||GT|}$ factor, is necessary in order to account for the fact that the cardinalities of $C$ and $GT$ are in general different (Cover and Thomas, 1991). Higher values of *NMI* mean better clustering quality. *NMI* is designed for hard clustering.

The normalized complementary entropy (Strehl, 2002, page 108) is a modified version of entropy which ranges in the interval $[0,1]$. The entropy of a cluster $c_j \in C$ is

$$E_j = \sum_{k=1}^{|GT|} -r_{k,j}\log r_{k,j}$$

where $r_{k,j}$ is the recall of cluster $c_k$ for class $GT_j$. To normalize the entropy we divide $E_j$ for $\log|GT|$. To evaluate the entropy for the whole clustering we can now sum the contribution of each cluster. Then we compute the complement of the entropy to make it coherent with the above described measures so that higher values mean better quality. The final formula for the normalized complementary entropy is the following:

$$NCE(C,GT) = 1 - \frac{1}{\log|GT|}\sum_{j\in 1}^{|C|} E_j$$

Table 4: Clustering performance when using the original pages, the proposed template removal algorithm with $t = 8, 16$, and the removal of the manually extracted template.

| | PI: punto-informatico.it | | | |
|---|---|---|---|---|
| | Original | Our | | Baseline |
| | | $t = 8$ | $t = 16$ | |
| F-measure | 0.234 | 0.260 | **0.276** | 0.274 |
| Purity | 0.249 | 0.261 | 0.272 | **0.279** |
| NMI | 0.040 | 0.044 | **0.059** | 0.058 6 |
| NCE | 0.064 | 0.068 | **0.084** | 0.083 |
| | Ansa: ansa.it | | | |
| | Original | Our | | Baseline |
| | | $t = 8$ | $t = 16$ | |
| F-measure | 0.387 | 0.445 | **0.449** | 0.422 |
| Purity | 0.397 | 0.466 | **0.471** | 0.452 |
| NMI | 0.346 | **0.425** | 0.384 | 0.346 |
| NCE | 0.383 | 0.388 | **0.421** | 0.383 |

Table 4 reports an evaluation of the clustering quality for the datasets PI and Ansa in different cases: when the template is not removed (column *Original* in the tables), when the template is identified and removed by the template extraction algorithm (with different values of the parameter $t$), and when we remove the manually extracted template. For each measure, we highlighted the case reporting the highest performance.

A first important observation is that, the experiments confirm how template removal is an effective tool to improve the quality of Web information retrieval tasks. Another important observation is that these experiments confirm how setting $t = 16$ in the extraction algorithm is, in general, a good choice independently of the document corpus characteristics.

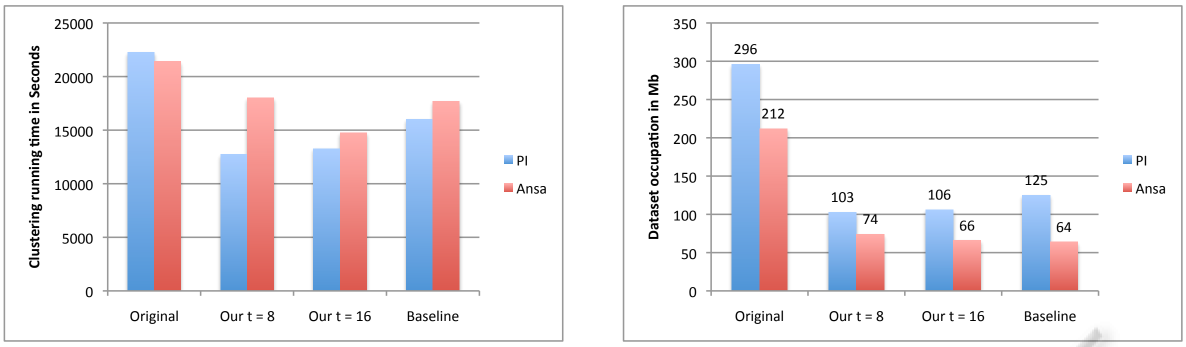
Hence, even if it works quite well in practice, the removal of the manually extracted template is not the best possible choice for improving the clustering quality. In fact, for this task the perfect identification of the template is not the main issue. The experiments show that the automatic template extraction algorithm consistently provides a better result in the clustering task when compared to the removal of the manually extracted template.

Figure 3 shows the benefits of Web template removal for clustering in terms of the reduction of the corpus size, and of the clustering running time. In particular, figure 3 (a) reports the clustering running time for PI and Ansa. The experiments confirm that Web template removal has consistently positive effects on the clustering time. The extraction algorithm makes clustering faster than that obtained by the manual Web template removal. In the case of PI The proposed algorithm can contribute to save up to 50% of the clustering time (compared to the clustering of the original dataset). In the case of Ansa the amount of saved time depends on the choice of $t$. Figure 3 (a) shows that, setting $t = 16$, the saved time is still about 40%.

Figure 3 (b) shows that the extraction algorithm reduces the amount of used space to store the document corpus of a factor 3. Note that for the PI dataset the algorithm reduces the corpus size more than the removal of the manually extracted template. This is due to the fact that, in this case, the extraction algorithm has identified snippets of HTML, which are repeated in the most of pages, even if they are not part of the Web template (i.e. tags commonly used for article formatting).

## 4 CONCLUSIONS

The paper describes a template extraction technique that exploits a sequence alignment algorithm to detect the HTML structure that is shared by the pages



(a) Clustering running time (b) Dataset size in Mb

Figure 3: Time and space requirements of the clustering algorithm for the datasets PI and Ansa before and after template removal.

from a target site. The algorithm is able to yield good retrieval performance given just a small number of pages from the site. It is also shown that the removal of the template extracted by the algorithm can significantly improve the performance of a Web page clustering application. Future work will concern the improvement of the alignment step by studying different solutions for assigning the similarity matrix on the basis of different features of the compared elements.

## ACKNOWLEDGEMENTS

This work was partially supported by the Italian Registry of the ccTLD ".it" Registro.it.

## REFERENCES

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

Chakrabarti, D., Kumar, R., and Punera, K. (2007). Page-level template detection via isotonic smoothing. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 61–70, New York, NY, USA. ACM.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. John Wiley & Sons, New York, US.

Debnath, S., Mitra, P., and Giles, C. L. (2005). Automatic extraction of informative blocks from webpages. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1722–1726, New York, NY, USA. ACM.

Geraci, F., Pellegrini, M., Maggini, M., and Sebastiani, F. (2008). Cluster generation and labelling for web snippets: A fast, accurate hierarchical solution. *Internet Matematics*, 3(4):413–443.

Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2/3):293–306.

Kovacevic, M., Diligenti, M., Gori, M., and Milutinovic, V. (2002). Recognition of common areas in a web page using visual information: a possible application in a page classification. In *Proceedings of the IEEE International Conference on Data Mining*, ICDM '02.

Needlemana, S. B. and Wunscha, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Song, R., Liu, H., Wen, J.-R., and Ma, W.-Y. (2004). Learning block importance models for web pages. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 203–211, New York, NY, USA. ACM.

Strehl, A. (2002). *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, University of Texas, Austin, US.

Wong, T.-L. and Lam, W. (2007). Adapting web information extraction knowledge via mining site-invariant and site-dependent features. *ACM Trans. Internet Technol.*, 7.

Yi, L., Liu, B., and Li, X. (2003). Eliminating noisy information in web pages for data mining. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 296–305.