A Survey On Various Web Template Detection And Extraction Methods

Neethu Mary Varghese, Tenny Thomas Soman

Abstract: In today's digital world, reliance on the World Wide Web as a source of information is extensive. Users increasingly rely on web based search engines to provide accurate search results on a wide range of topics that interest them. The search engines, in turn parse the vast repository of web pages searching for relevant information. However, majority of web portals are designed using web templates, which are designed to provide consistent look and feel to end users. The presence of these templates however can influence search results leading to inaccurate results being delivered to the users. Therefore to improve the accuracy and reliability of search results, identification and removal of web templates from the actual content is essential. A wide range of approaches are commonly employed to achieve this, and this paper focuses on the study of the various approaches of template detection and extraction that can be applied across homogenous as well as heterogeneous web pages.

Index Terms: Cluster, Homogeneous web page, Heterogeneous web page, Page-level detection, Search engine, Site-level detection, Template Detection, Template Extraction.

1 INTRODUCTION

In the current digital age, the World Wide Web (WWW) is the most widely accessed provider of data and information Web portals and applications are comprised of web pages which are increasingly being generated using templates, leading to more than 50 percent of the data within a web page, on an average comprising of template information. With the increasing reliance on enterprise tools and accelerators for building web applications, this figure is set to go further up. Templates, as the name suggests are a framework on which actual content is built. From an end user point of view, templates provide a consistent look and feel within a particular web site thus enhancing the user experience and making it easier to navigate around. However, on the other hand, the presence of these templates in large proportions in web pages can compromise the performance and accuracy of search results as the search engines often parse the non-relevant template information instead of actual content which can lead to users getting directed to non relevant web pages. This underscores the need to ensure that templates are effectively removed from web pages prior to processing by search engines to ensure the most accurate and relevant results are provided in response to user queries. There are different methods available for template detection and extraction. This paper mainly focuses on a brief comparison between the various template detection and extraction methods. Many of the previous methods [2], [3], [5], [6], [10] were based on the assumption that all the web pages are of homogeneous type. The use of factors like Tree-edit distance [2], [4] is very much expensive. Many existing approaches [3], [6], [8] use word frequency to find out the similarities between pages. A threshold value [7] of text frequency in pages is. Template detection can also be performed based on a single page [9] instead of using a set of pages. Another latest approach [1] considers heterogeneous web pages as input and it detects templates based on two decisive factors such as frequency as well as a principle called MDL (Minimum Description Length). The paper is organized as follows:: Section 2 includes a brief study on various template detection and extraction methods. In section 3, a comparison of the different template detection and extraction methodologies are included and Section 4 includes a brief conclusion.

2 VARIOUS TEMPLATE DETECTION AND EXTRACTION METHODS

Template detection and extraction methods are mainly classified as: Site-level method and Page-level method. The Site-level methods detect templates based on several pages from a site. A set of sample pages will be collected as input and templates are detected based on various factors like similarity criteria or a threshold value depending on the technique used. The Page-level methods detect templates based on a single page. A page is taken as input and decision on templates is made based on a certain similarity criteria or a threshold value based on the technique used. A method proposed by Chulyun Kim and Kyuseok Shim [1] is a site-level type of template detection and extraction in which templates are found out from a set of heterogeneous web pages from a site. The method works automatically with the help of algorithms and it performs a type of grouping known as clustering based on similarities existing in the input pages. It uses a principle known as Minimum Description Length (MDL) for detecting templates and calculates a value termed as MDLcost to identify the best cluster. The method proposes an algorithm known as TEXT-MDL and extracts the detected templates. The method proposed by K.Viera et al. [2] is a Site-level detection method in which templates are detected automatically based on several pages of a web site. The method is based on the assumption that the input pages are of homogeneous nature. In addition to detection, removal or

- *Neethu Mary Varghese has completed her Masters in Computer Science and is working as an Assistant Professor in Mar Baselios Institute of Technology and Science, India. PH-+919895048869.*
  E-mail: neethu02@gmail.com
- *Tenny Thomas Soman is currently working as a Business Intelligence Consultant at Capgemini Financial Services Ltd, UK, PH-+447448577466.*
  E-mail: tennytts@gmail.com

extraction of templates is also done afterwards. The method works in two steps: (1)Detect templates from a set of input pages (2)Remove the detected templates. Input pages are represented in the form of Document Object Model (DOM) trees. A minimum cost mapping in terms of a factor called Tree-edit distance is found out for finding similarities between pages. Tree-edit distance is defined as the cost associated with the minimal number of operations needed to convert one tree to another, when two trees are considered. Identical nodes and sub-trees containing such nodes are found out from the tree structure of pages. If a particular sub-tree is found in two input pages under consideration, it is considered as a template. The method proposes 4 algorithms such as ExtractSubTree, RTDM-TD, retrieveTemplate and findTemplate. The findTemplate algorithm finds and removes 2 pages from the set of input pages. The operation is performed in a random fashion. The ExtractSubTree algorithm extracts sub-trees that are similar from the removed pages. The RTDM-TD algorithm takes two trees as input and gives a cost matrix and a backtracking matrix as result. The retrieveTemplate algorithm takes the output of RTDM-TD as input and gives template nodes as output. In this method, template removal is inexpensive, once it is detected. Another Template detection problem studied by Z.Bar-Yossef and S.Rajagopalan [3] is a Site-level detection method that focuses on detecting templates from a set of homogeneous pages. The paper focuses on finding a solution to the proposed problem based on counting the number of items that occur often. If a particular item is found to be repeated many times it is counted as a template. Two main algorithms are proposed in this method such as Local Template detection algorithm and Global Template detection algorithm. Local Template detection algorithm works with small input sets. Global Template detection algorithm works with large input sets. Another type of template extraction proposed by M.de Castro Reis et al. [4] is a Site-level type of extraction that extracts data in the web automatically. The method mainly deals with detecting and extracting data like web news that are present in many web sites. A type of grouping concept called clustering is used which uses a factor known as Tree-edit distance for detection and extraction. This concept employs similarities in the underlying structure of target pages in the target site is considered for the clustering process. Input pages with similarity in structure are grouped together and clusters are formed. Tree-edit distance is used to evaluate similarities in the structure of pages. The method mainly focuses on finding the least cost mapping between the trees under consideration. There are 2 main tasks involved in this approach such as collecting required pages by means of crawlers and extraction of web news from the crawled pages. Four steps are involved in the extraction process: (1) clustering web pages (2) Generation of patterns for extraction (3) matching of data (4) labeling of data. The first step involves making use of a type of clustering known as hierarchical clustering to form clusters of pages. It also uses a threshold value to determine whether two clusters can be merged or not. Finally, several clusters will be formed which conforms to same template structure. The next step generates extraction patterns of nodes which are special type of trees that takes each page in the cluster as input. Next step involves matching the node extraction patterns with target pages and the contents are extracted. Last step involves finding the required information like title and body of web news from the contents extracted in the previous step. Yet another closely related technique proposed by L.Yi et al. [5] is also a Site-level method. It deals with removing noisy information from web pages. Noisy information includes advertisements, common links etc. The technique is mainly based on an observation that blocks containing noisy data will have some common data and styles of presentation, while blocks containing actual data will be different in their data and styles. Therefore a kind of tree called style tree is proposed to represent the actual data and common presentation styles of pages in a web site. For a site, site style tree(SST) is built by using sample pages of that site. Afterwards certain portions of site style tree are identified as noises and certain portions as main contents. To remove noises from any web page, the corresponding page can be mapped to site style tree. Yet another Site-level method proposed by A.Arasu and H.Garcia-Moilina [6] deals with extracting data from web pages automatically. It is also based on the assumption that the input pages are of homogeneous type that conform to a common template structure. The technique derives templates from the web pages that conform to a common template as a first step and thereafter extracts data from the derived templates. An algorithm called as EXALG(Extraction Algorithm) is proposed in this paper which involves mainly 2 modules such as Equivalence class generation module and Analysis module. Equivalence class generation module generates equivalence classes as result. Analysis module takes equivalence classes as input and generates templates and data as output. There is another technique proposed by L.Ma et al. [7] which is also a Site-level type of template detection and extraction. The method tries to detect templates in web pages and thereafter extracts text data so that it is free of templates. Table tags are used by web pages to differentiate template from text. A type of structure called as table text chunk is used in this method which can be defined as a group of terms that exist between a pair of table tags. As a first step, when a table tag and its corresponding end tag is found, the table text chunk encountered is placed in a data structure known as text chunk map along with its frequency of occurrence. When each page is processed, frequency is increased or kept constant. The text chunks with frequency of occurrence greater than a certain threshold value is identified as template. Other text chunks are identified as non-templates. There is another method proposed by Liang Chen et al. [8] which is also a Site-level type of template detection and extraction in which templates are detected and removed while search engines build indices. The template detection and removal are combined with the index building process and it is carried out in two stages. In the first stage, blocks are formed from web pages and blocks with similar structure are grouped together to form clusters. In the second stage, blocks with similar contents are identified as templates and they are extracted. During the index building process of search engine, frequency of words and their positions are computed for each and every block in order to find out the similarity between contents. Blocks with similar contents and similar structure are identified as templates. The method proposed by Yu Wang et al. [9] is a page-level type of template detection that detects templates on a page by page basis. The method proposes a framework that detects templates based on historical information stored about web pages. As soon as a page is collected, it is passed through the process of template detection. Detection is done based on the repetition of text portions termed as text segments in pages. A special type of data structure is used to store text segments and their number of occurrences. Pages are represented in the form of Document Object Model (DOM) trees. Two text segments are considered to be equal, if their contents are equal and also their paths in the DOM trees are same. It also checks whether each text segment is already present in the data structure or not. If already present, its number of occurrence is incremented by 1. Template ratio is also calculated which is the ratio of total length of all template segments to total length of all text segments. If the ratio is greater than a certain threshold value, that block is detected as template block. Yet another method proposed by Sandip Debnath et al. [10] is also a Site-level type of detection method that distinguishes between relevant data and non-contents or templates and thereafter extracts contents. The extraction is done automatically without any human intervention. The method proposes two algorithms known as ContentExtractor and FeatureExtractor that extracts content blocks. The ContentExtractor algorithm distinguishes between content blocks and non-content blocks based on repetition of same blocks in several pages. The FeatureExtractor algorithm extracts contents based on features supplied externally. The algorithms deal with web pages having similar underlying template structure. The inputs to these algorithms are a set of homogeneous web pages and output consists of content blocks. Content blocks and non-content blocks are distinguished using a factor known as block document frequency which is the frequency of blocks in documents. If the frequency is high, it is identified as non-content block or template block.

3 COMPARISON OF VARIOUS TEMPLATE DETECTION AND EXTRACTION METHODOLOGIES

TABLE 1
COMPARISON OF TEMPLATE DETECTION AND EXTRACTION METHODS

Method	Approach	Underlying technique	Input	Output	Factors affecting detection and extraction	Classification	cost
Template detection and extraction[1]	automatic	clustering	Heterogeneous web pages	Template	MDLcost	Site-level	Not much expensive
Template detection and removal[2]	automatic	Mappings between DOM trees of distinct pages	Homogeneous web pages	Template	Tree-edit distance	Site-level	Not much expensive
Template detection[3]	automatic	Counting frequent item sets	Homogeneous web pages	Detected template	Frequency of items	Site-level	Not much expensive
Data extraction[4]	automatic	Hierarchical clustering	Heterogeneous web pages	Data	Tree-edit distance	Site-level	expensive
Template detection and extraction[5]	Semi-automatic	Building SST(site style tree)for a site	Homogeneous web pages	noise	Information based measures	Site-level	Not much expensive
Data extraction[6]	automatic	Finding Frequency of words	Homogeneous web pages	Template, data	Frequency of words	Site-level	Not much expensive
Text extraction and making it template-free[7]	automatic	Frequency of occurrence of text	Heterogeneous web pages	Template-free text	Threshold value for document frequency of text	Site-level	Not much expensive
Template detection and extraction[8]	automatic	Segmentation and clustering	Heterogeneous web pages	Template	Frequency of words and position of words	Site-level	Not much expensive
Template detection[9]	Semi-automatic	Segmentation	Single web page	Detected Template	Text segment repetition	Page-level	Not much expensive
Content extraction[10]	automatic	Partitioning into blocks	Homogeneous web pages	Content blocks	block repetition	Site-level	Not much expensive

4 CONCLUSION

Templates in the form of advertisements, links etc deviate users from their actual intention and hence such irrelevant matters are to be removed from web pages. As a result, search engines will be able to retrieve the most suitable pages for users, thus increasing the importance of real contents. A brief comparison of the various approaches in detecting and extracting templates is done in this paper. Although each method has its own advantages and disadvantages, it is recommended that the scope of future analysis includes a combination of multiple techniques to establish an ideal solution that is effective in terms of both cost and performance.

REFERENCES

[1] Chulyun Kim and Kyuseok Shim, “Text:Automatic Template Extraction from Heterogeneous Web Pages,” IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 4, April 2011.

[2] K.Vieira, A.S. da Silva, N.Pinto, E.S. de Moura, J.M.B. Cavalcanti and J.Friere, “A Fast and Robust Method for Web Page Template Detection and Removal,” Proc.15th ACM Int’l Conf. Information and Knowledge Management(CIKM), 2006.

[3] Z.Bar-Yossef and S. Rajagopalan, “Template Detection via Data Mining and its Applications,” Proc.11th Int’l Conf. World Wide Web(WWW), 2002.

[4] M.de Castro Reis, P.B.Golgher, A.S. da Silva and A.H.F Laender, “Automatic Web News Extraction Using Tree Edit Distance,” Proc.13th Int’l Conf. World Wide Web(WWW), 2004.

[5] L.Yi, B.Liu and X.Li , “Eliminating noisy information in Web Pages for Data Mining,” In Proceedings of the International ACM Conference on Knowledge Discovery and Data Mining, 2003.

[6] A.Arasu and H.Garcia-Molina, “Extracting Structured Data from Web Pages,” Proc.ACM SIGMOD, 2003.

[7] L.Ma, N.Goharian, A.Chowdhury and M.Chung, “Extracting Unstructured Data from Template Generated Web Documents,” Proc. CIKM, pp 512-515, 2003.

[8] Liang Chen, Shaozhi Ye, Xing Li, “Template Detection for large scale search engines,” Proc.ACM Symposium, pp 1094-1098, 2006.

[9] Yu Wang, Bingxing Fang, Xueqi Cheng, Li Guo, Hongvo Xu, “Incremental Web Page Template Detection,” Proc.17th Int’l Conf. World Wide Web(WWW), pp 1247-1248, 2008.

[10] Sandip Debnath, Prasenjit Mitra, C.Lee Giles, “Automatic Extraction of Informative Blocks from Web Pages,” Proc.ACM Symposium, pp 1722-1726, 2005.