International Journal of Electrical, Electronics and Computer Engineering **4**(1): 125-131(2015)

ISSN No. (Online): 2277-2626

Template Extraction from Heterogeneous Web Pages

Harshal H. Kulkarni and Mrs. Manasi K. Kulkarni
Asst. Professor, (PES's MCOE)
Department of Computer Science, Pune University, (MS), INDIA

(Corresponding author: Harshal H. Kulkarni)
(Received 23 April, 2015 Accepted 28 June, 2015)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: The meaning of template in web application is that it is structural information of any web site. So that templates are created for many web sites to increase the productivity and search time of web pages. The templates have consistent structure so that users can easily access the information on the web sites. In some times templates are considered harmful because of its irrelevant words in the template, so it will affect on the performance of web applications. Thus, template detection and extraction techniques have received a lot of attention to improve the performance of search engines, web application, clustering and classification of web documents. The objective of this paper is to cluster the web documents based on the similarity of underlying template structures in the documents so that the templates for each cluster are extracted simultaneously. While extracting the templates, we consider the web page structure along with its contents. To effectively manage an unknown number of clusters we propose an algorithm by using dice-coefficient method based Template Extraction.

Keywords: Template Extraction, HTML Documents, similarity, Dice-coefficient.

I. INTRODUCTION

The word "Template" is going to be used for defining structural information of different areas, such as web applications areas, biometric areas, digital System areas, Programming Languages. In the web application areas the templates show the structural information of any web sites, means they are nothing but the Master page of that web site. Normally a web template contains common elements of the web pages. The common elements of these template (i.e. master pages) pages are going to focus on their linked page. Examples of these elements are any college or school web sites, any business related web site, any online shopping web sites etc. [8] [9]. The following Fig.1 shows a simple example of the school web template. In this example, the website contains different elements. (1)About, (2) Meet the staff, (3) Principal's notes (4) Faculties, (5) Curriculum (6) Our Contacts. [8]. As we know the World Wide Web is huge data collection system, where huge number of data is going to extracted or submitted daily.

Fig. 1. Shows a simple example of the school web template.

We can access this data (information) with the help of web sites in terms of web pages. To access this huge data from the web sites we need to achieve the efficient access and less search time of the web pages, the web pages are published on many web sites with its common template. The templates provide users easy access to the information with its common contents and consistent web page structures.

When unknown templates are generated due to their irrelevant term then, they look harmful for search engine, web applications etc. because they degrade the accuracy and the performance of the system. Thus template detection and extraction techniques have received a lot of attention to improve the performance of the web applications such as data integration, search engines, classification of the web documents and so on [1][2][3]. The objective of this paper is to cluster the web documents based on the similarity of underlying template structures in the documents so that the templates for each cluster are extracted simultaneously. While extracting the templates, we consider the web page structure along with its contents. To effectively manage an unknown number of clusters we propose an algorithm by using dice-coefficient method based Template Extraction which generates the qualified clusters.

II. RELATED WORK

The "Template-Extraction" problem is based on Data Mining Concept. Initially template extraction problem is studied by Yossef & Rajagopalan [8]. In that, they use html tags of the web pages for template extraction system. But they analyzed that the word is also part of web pages so it can be used for template extraction. And this approach was implemented by Arasu and Garcia-Molina, where they observed that, for real pages, the words that are part of the template have a high correlation of occurrence in input pages with other words in the template. After that Reis, Golgher, A.S. da Silva, and A.H.F. Laender, introduced a Tree Edit Distance method and RTDM algorithm [5]. Sruthi Kamban, K.S, M.Sindhuja, they propose a method to cluster the web pages with the help of its URL and Document Object Model (DOM) clustering method [6]. Jorma Rissanen introduces Minimum Description Length (MDL) principle is how to estimate the shortest code length for the data, given a suggested model class [4]. Chulyun Kim and Kyuseok Shim give brief idea how to detect and extract the template from heterogeneous web pages.

A. Parsing the Web documents using HTML Parser and DOM Approach

This approach gives simple way to represent HTML Document in tree format [9]. In this model we can access and modify the style, structure and content of web document at run time. This approach is basically built for HTML or XML documents. The DOM also includes standard way to represent HTML or XML documents in tree structure.

First node of the tree always defines Document node and every HTML tag and text in the HTML element are treated as an element node and text nodes respectively. Every HTML attribute is an attribute node. Since all words are equally treated as defined, any type of node can be a part of a template. Consider following two figures where Fig. 2 is simple HTML document and Fig. 3 represent DOM tree of this HTML document.

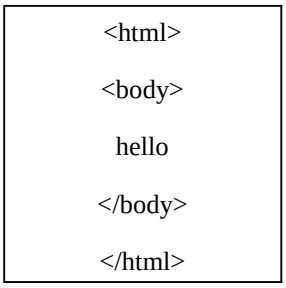

Fig. 2. Simple HTML Document d.

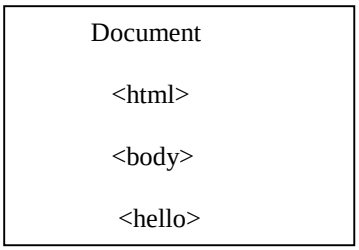

Fig. 3. DOM tree of d.

In a DOM tree, we can write a path of any node as follows: in Fig. 4 the path of a node "list" is "Document\<html>\<body> \ hello."

<html> <html>	<html>	<html>	
<body>	<body>	<body>	<body>
word	Local	Global	hello
 	 	 	</body>
</body>	List	List	</html>
</html>	</body>	</body>	

Fig. 4. Simple Web Documents (a) Document *d*1 (b) Document *d*2(c) Document *d*3 (d) Document *d*4.

B. Generation of Essential path

Essential paths are the path whose support value is greater than or equal to some threshold. Suppose given web documents collection D= {d1, d2, d3, d4}, depicted in Fig. 4 and in Table 1 represents their paths and supports.

Table 1: Documents paths and support values.

ID	Path	support
P	Document\<html>	4
P2	Document\<html>\<body>	4
P3	Document\<html>\<body>\<h1>	3
P4	Document\<html>\<body>\ 	3
P5	Document\<html>\<body>\<List>	3
P6	Document\<html>\<body>\<h1>\word	1
P7	Document\<html>\<body>\<h1>\Local	1
P8	Document\<html>\<body>\<h1>\Global	1

Consider P_d as a set of all paths in. Here we do not consider document node because it is available in all DOM tree so it is not include in p_d set. To reduce the similarity costs do not considering all the paths of document d_i, Instead of that select some *essential paths* from the set P_d. The *support* value of a path P_i is equal to number of documents from set D including path P_i. Set individual threshold td_i for each document. Set Minimum support value for the path p_i where p_i € p_d and set this minimum support value as a Threshold td_i for document. Notice that the threshold values of two different documents may be different. If any path p_i € p_d and support value of path p_i is greater than or equal to threshold value of document d_i then, path p_i is called as essential path of document d_i.

So the essential paths of the given four documents are,

d1= {p1, p2, p3, p4}, d2= {p1, p2, p3, p4, p5}

d3= {p1, p2, p3, p4, p5}, d4= {p1, p2}

C. Matrix representation of cluster

The next illustration is representation of clustering of web document and essential path in matrix form. Usually matrix representation involves a tradeoff among accuracy, the memory space occupied by the representation and the time required to obtain the representation. Note that in our experiment only one document can enter only in one cluster. We can draw matrix which gives the Relation between the documents (as discussed previously) and there essential paths in ME matrix, where each column & row in Document Essential path (ME) matrix represents the document and the essential paths of that document respectively. MT matrix represents clustering information with its path, where MD is representation of cluster with documents [2].

$$ME = \begin{array}{c|cccc} & d_1 & d2 & d3 & d_4 \\ P_1 & 1 & 1 & 1 & 1 \\ P_2 & 1 & 1 & 1 & 1 \\ P_3 & 1 & 1 & 1 & 0 \\ P_4 & 1 & 1 & 1 & 0 \\ P_5 & 0 & 1 & 1 & 0 \\ P_6 & 0 & 0 & 0 & 0 \\ P_7 & 0 & 0 & 0 & 0 \\ P_8 & 0 & 0 & 0 & 0 \end{array}$$

D. Minimum Description Length Principle

The MDL principle states that "the best model inferred from a given set of data is the one which minimizes the sum of 1) the length of the model, in bits, and 2) the length of encoding of the data, in bits, when described with the help of the model" [4] . The MDL costs of a clustering model C and a matrix M are denoted as $L(C)$ and $L(M)$, respectively. Considering the values in a matrix as a random variable X, p_r (1) and p_r (−1) are the probabilities of 1 s and -1 s in the matrix and pr (0) is that of zeros. Then, the entropy H(X) of the random variable X,

$H(X) = \sum -p_r(x) \log 2pr(x)$ and
$x \in \{1,0,-1\}$

$L(M) = M.H(X)$

The MDL cost of clustering Model C is calculated as,

$L(C) = L(M_T) + L(M_D) + L(M_\Delta)$

E. TEXT-MDL (clustering with MDL cost)

This model describes the clustering web documents with MDL cost. Here set of documents D is input and qualified generated cluster is an output. When we merged a pair of clusters, the MDL cost of the clustering model can be reduced or increased. To find the Get Best Pair, pair of clusters is find out whose reduction of the MDL cost is maximal in each step of merging and the pair is repeatedly merged until any reduction is not possible.

In order to calculate the MDL cost when each possible pair of clusters is merged, the procedure GetMDLCost (ci, cj, C), where ci and cj are a pair to be merged and C is the current clustering, is called in GetBestPair and C is updated by merging the best pair of clusters [2]. When we get first pair with the MDL cost reduction affects to remaining clusters in mode C, therefore GetBestPair should recalculate the MDL cost between the first best pair and all other clusters in C. It will increase the time complexity of algorithm. Hence it is not good practice to use TEXT-MDL algorithm with large number of web documents [2].

F. TEXT-MAX (clustering with min hash and Jaccord coefficient)

The evaluation of this algorithm is depending on signature of document and Jaccord coefficient. The input parameter for this model is documents set D and output is cluster set C. The signature of document is calculated by assigning random rank to every element of the document. With the help of signature we can found the number of essential paths which are present in both the documents. When we get the same signature value of cluster then merge this cluster in one pair. By using Jaccord coefficient method we can reduce the search space for getting GetInitBestPair [10] .The complexity of GetInitBestPair and GetBestPair depend on the number of clusters selected in maximal Jaccard's coefficient [2].

G. Template extraction with cosine similarity

This is another approach of extracting the template from heterogeneous web pages. In this approach we can calculate the similarity between two web pages. Here we considered the similarity values in the form of 1's and 0's. If similarity of both the document is equal to one then documents are similar and grouped them in to one cluster.

The cosine similarity formula is,

Cosine similarity of two documents d1, d2,

Cosine= (d1*d2) / |d1|*|d2|;

III. PROPOSED WORK

In this section we propose an algorithm using dice-coefficient clustering method to cluster the heterogeneous documents. The basic concept of dice-coefficient is to find out the word or string similarity between the documents. The concept of word or string similarity is to find out the likeness of their meaning, semantic content of the set of the documents or terms within terms list.

In the dice-coefficient the strings are converted in to bigrams form. After converting it we check how many common strings are finding between them and use dice-coefficient formula to find out similarity between two strings or words. The dice-coefficient values are taken between 1's or 0's. If the two words or strings are perfectly matched with each other their dice-coefficient value is said to be 1. Example 1 illustrates calculation of dice-coefficient.

Example1.

Suppose we have two strings s1= "christ" and s2="christos".

We convert these strings into bigrams. The bigrams of the two strings are,

(i) "christ" = {'ch','hr','is','st'}

(ii) "christos" = { 'ch','hr','is','st','to','os'}

By using dice-coefficient formula,

Dice-coefficient= (2* C) / S1+S2;

Where,

(i) C is no. of character (common) bigrams found in both strings.

(ii) S1 is no. of unique bigrams in string s1.

(iii) S2 is no. of unique bigrams in string s2.

From the above example the values of,

C = 5, S1 = 5, S2 = 6. Put these values in the above equation. We get dice-coefficient value,

Dice-coefficient= (2*5) / (5+6),

Dice-coefficient=0.90909091.

The input to the algorithm is set of web documents (as depicted in Fig. 4) D and output is the clustered templates C. Initially one document can enter in one cluster only. To simplify the clustering calculation the web document is represented in DOM tree format [6]. After representation the essential path of every document is generated and with the help of essential path we can draw ME matrix (initially discussed in section B and C) [2] for dice-coefficient value. Here we initially Clustered the web documents whose essential path set is similar. Means the documents whose essential path is similar grouped them in to one cluster before applying dice-coefficient method. With the help of this idea we can reduce the execution time and further calculations. And remaining calculation is done for remaining web documents. To calculate similarity between document d1 & d2 take the first and second column of ME matrix and then find out the similarity as calculated in Example1. Similarly calculation process will repeat for all the clusters present in set C.

Example 2.

Consider above four documents (Fig. 4) for the experiments. Let Consider,

C= {(c1=d1), (c2=d2), (c3=d3), (c4=d4)},

Then we have to find out similarity calculation of document d1 with the other documents as in the pairs, {(d1-d2), (d1-d3), (d1-d4)}, similarly of document d1, {(d2-d1), (d2-d3), (d2-d4)}, similarly of document d2, {(d3-d1), (d3-d2), (d3-d4)}, similarly of document d3, {(d4-d1), (d4-d2), (d4-d3)},similarity of document d4.

As we mentioned early the documents having similar essential path directly put in to one cluster. So with the reference of ME matrix, the document d2 and d3 are having similar essential path so put it in cluster. Initially put one document into single cluster to select best pair among them. For that threshold value is calculated. For calculation of threshold value no need of external input parameter, it can be calculated as average of dice_coefficient_values. After that select those pair whose dice_coefficient_value is greater than or equal to threshold value. Arrange all the selected pairs with their maximum dice-coefficient value. After successful calculation of dice_coefficient_values, the clustering process start with the set (*similarity_set*) of pairs such that (ci, cj, dice_coefficient_value). The first pair in this is the best pair to merge in which documents di and dj & is approximately similar to each other. Documents di and dj, will form the new cluster ci ; Note here in the cluster ci only documents are included, to generate the template path one has to follow the template path generation step. After merging the documents di & dj in cluster c*i* remove all the entries of di & dj from the *similarity set.* Repeat the clustering and removing steps till there is a not a single entry in *similarity_set*. At last cluster those documents which are near to each other. Few documents are such type which are not related to any other document hence they are not the part of *similarity_set* such documents are consider as a single template & template path for that can be only the essential paths of same document.

1. C= {c1, c2, c3, … , cn} with ci={E(di),{di}}
2. Meage all cluster with the same E (di).
3. For each ci in C do {
4. ctemp =C-ci.
5. for each cj in ctemp{
6. (ci, cj, dice-coefficient) = calculate similarity value for each pair ci and cj.
7 .similarity_set= {ci, cj, dice-coefficient};
8. count++;
9.}
10.}
11. //set threshold value tdi,

$$tdi = \frac{\sum dice\_coefficient\_vale}{count}$$

12. Discards all the entries from similarity set whose dice_coefficient_values less than tdi.
13. Arrange similarly _set in descending orders.
14. While similarity_set is not empty {
15. pick up the ci and cj one after from similary_set
16. ci_temp=ci ; cj_temp=cj;
17. ck=ci U cj;
18. Remove all the entries of ci and cj from similarity_set.
19. Similarity_set=similarity_set – {ci,cj}
20. C=C-{ci,cj}-ck;
21.}

Fig. 5. Dice-coefficient Algorithm.

Fig. 5 shows an algorithm to cluster the web documents using dice-coefficient method. Initially at line 1 each cluster contains the single document and essential path set. Merge all clusters having the same essential path set E (di). After that from line 3 to 9 calculate the similarity between all the cluster pairs and insert these pairs into *similarity set by calculating dice_coefficient_value*. To choose the best cluster pair to merge, set the threshold value and the cluster pair whose dice_coefficient_ value is less then are discarded from the *similarity Set*. At last from line 13 till 20 select the pairs of & one after other from *similarity set* & merges them. Discard all the entries of & from similarity set. Update the cluster set C. = {ci cj}-ck;

IV. RESULT AND DISCUSSION

The aim of the project is to extract a template from heterogeneous web pages. Different extracting algorithms are implemented like TEXT-MDL, TEXT-MAX, cosine similarity and dice-coefficient. Result analysis is carried out in terms of time required to extract template and number of compact clusters are generated by previous implemented algorithms. After successful execution of all four algorithms, here we conclude that dice_coefficient generates better clustering output as compare to other, and there is slight difference in execution of time. All experiments were performed on a Pentium(R) Dual-Core 2.00 GHz machine with 2 GB of main memory, running Windows XP operating system. All algorithms were implemented in JAVA with JRE version1.7.0.

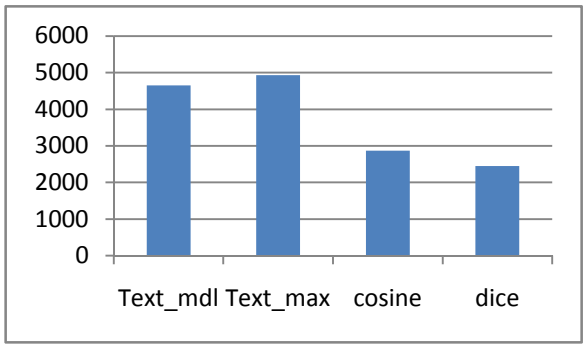

Fig. 6. Bar chart for Time Analysis.

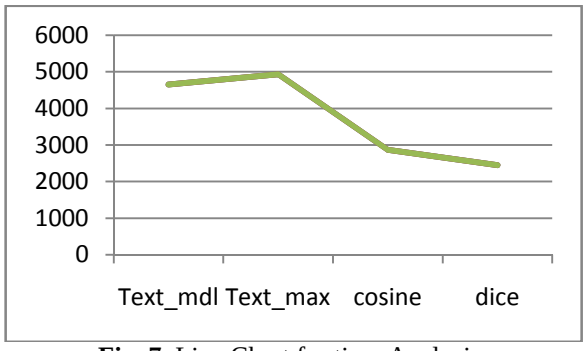

Fig. 7. Line Chart for time Analysis.

Fig. 6 and 7 shows the graphical representation of the experimental results. To conform the effectiveness of algorithm using dice_coefficient method the heterogeneous web pages are collected from sophisticated web sites like online shopping sites, some government site etc. same set of web pages given as input to TEXT_MDL, TEXT_MAX, Cosine Similarity and dice_coefficient. Dice_coefficient algorithm is more effective than remaining other algorithms in terms of Evaluation of time and compactness of clustering.

V. CONCLUSION

Automatic template extraction system combines the contents of heterogeneous web pages into templates according to structure similarity of documents with its contents. So that required information can be retrieved with minimum search space. Extraction is dependent on the underlying structure of web documents. While extracting the template from collection of documents this technique not only considered the tags available in web designing but also considers contents as a part of the template along with tags. Web documents are represented into DOM structure to simplify the further clustering calculation. Employed MDL principle manages the unknown number of clusters. And also use dice-coefficient formula to calculate similarity between the web documents. The experimental result shows that dice-coefficient method consumes less time as compared to other methods.

VI. FUTURE SCOPE

The basic directions are encountered for future work. More attention should be given on design and structure of templates such as considering attributes of tags, images etc.

ACKNOWLEDGMENT

I wish to express my sincere thanks and deep gratitude towards Dr[Mrs.]. K. R. Joshi [Principal Pesmcoe] and my guide Prof. M. K. Kulkarni for her guidance, valuable suggestions and constant encouragement in all phases. I am highly indebted to her help in solving my difficulties which came across whole paper work. Finally i extend my sincere thanks to respected head of the department Prof. S.A. Itkar And Prof. D. V. Gore [P. Gco-Ordinator] and all the staff members for their kind support and encouragement for this paper. I wish to thank my husband and for his unconditional love and support.

REFERENCES

[1]. Arasu and H. Garcia-Molina, (2003). "Extracting Structured Data from Web Pages", *Proc. ACM SIGMOD, 2003*.

[2]. Chulyun Kim and Kyuseok Shim, (2011). "TEXT: Automatic Template Extraction from Heterogeneous Web Pages", *Ieee Transactions On Knowledge And Data Engineering*, Vol. **23**, NO. 4, APRIL 2011.

[3]. F. Pan, X. Zhang, and W. Wang, Crd. (2008). "Fast CoClustering on Large Data Sets Utilizing Sampling-Based Matrix Decomposition". *Proc. ACMSIGMOD,* 2008.

[4]. Jorma Rissanen, "Fisher Information, Stochastic Complexity and Universal Modeling", IBM Research Division, Almaden research center , 650 Harry Road, San Jose Ca 95120-6099.

[5]. de Castro Reis, P.B. Golgher, A.S. da Silva, and A.H.F. Laender, (2004). "Automatic Web News Extraction Using Tree Edit Distance", *Proc.13th Int'1 conf. World wide web* (www), 2004.

[6]. Sruthi Kamban, K.S, M. Sindhuja, (2013). Extraction of html document from heterogeneous web pages for clustering Tech", *International Journal of Engineering and Computer Science*, ISSN: 2319-7242, 4 April 2013.

[7]. Valter Crescenzi, Paolo Merialdo, Paolo Missier, (2005). Clustering Web Pages Based on Their Structure", *Data and knowledge Eng*, Vol. **54**, pp.279, 299, 2005.

[8]. Z. Bar-Yossef and S. Rajagopalan, (2007). "Template Detection via Data Mining and Its Applications," Proc. 11th Int'l Conf. World Wide Web (WWW), 2002.

[9]. S. Zheng, D. Wu, R. Song, and J.-R. Wen, "Joint Optimization of Wrapper Generation and Template Detection," Proc. ACM.

[10]. SIGKDD, 2007.Z. Chen, F. Korn, N. Koudas, and S. Muithukrishnan,(2000) "Selectivity Estimation for Boolean Queries," Proc. ACM SIGMOD-SIGACTSIGART Symp. *Principles of Database Systems (PODS)*, 2000.