AN EFFICIENT APPROACH FOR TEMPLATE EXTRACTION

Pravallika.CH [1]
Department of CSE, ANURAG GROUP OF INSTITUTION (CVSR),
Venkatapur (V), Ghatkesar (M), Andhra Pradesh, India

Swapna Goud.N [2]
Associate professor, Department of CSE, ANURAG GROUP OF INSTITUTION (CVSR),
Venkatapur (V), Ghatkesar (M), Andhra Pradesh, India

Vishnu Murthy.G [3]
Professor, Department of CSE, ANURAG GROUP OF INSTITUTION(CVSR),
Venkatapur (V), Ghatkesar (M), Andhra Pradesh, India

***ABSTRACT* ---The World Wide Web is a vast and rapidly growing source of useful information which is used to publish and access the information on the Internet. It uses different templates with contents for providing easy access for readers. But, for search engine detecting the template and displaying the content to the users is a major task in retrieval of web pages from the web. The templates are considered harmful because they compromise the performance of clustering and classification of the web pages. In this paper, we present novel algorithm for extracting templates from web documents which are generated from heterogeneous template structures. In the proposed, we are clustering the web documents based on the similarity in the template structure so that the template for each cluster is extracted simultaneously. The resultant clusters will be given as input to the Roadrunner system, which is used to extract information from template web pages.**

Keywords: Template extraction, Clustering web pages, MDL principle.

I. INTRODUCTION

The World Wide Web is a vast and rapidly growing source of useful information which is used to publish and access the information on the Internet. In which the most of this information is in the form of unstructured text, which makes hard to query the information required. Templates are wildly used in Web sites development which provides the managed web sites with an easy to manage uniform look and feel for the representation of web page information more clearly. Finding the template for a given set of Web pages could be very important and useful for many applications like Web page classification and monitoring content and structure changes and clustering of Web pages. Different templates are used for visualizing the content of web page. Server need to detect the template of a web page before publishing it when requested by the user, which has given a lot of attention to template detection in recent times. Several algorithms are developed to detect this template structures automatically in order to identify and extract contents of a documents. Clustering is done for automatic template detection and extraction and group the web pages based on the similarity of the documents. In present system a agglomerative hierarchical clustering technique was proposed to cluster the web pages according to their HTML tree structures. A basic measure-edit distance is calculated by comparing two DOM trees, which shows how similar the web pages are. The Document Object Model (DOM) is used for organizing data in web pages in a tree format. XML presents this data as documents, and the DOM may be used to manage this data by using data nodes information.

By evaluating the structural similarities between pages in a target site we are able to perform tasks such as grouping together pages as cluster with similar structure. However, clustering is very expensive and difficult in comparisons with tree-related distance measures. For instance, tree-edit distance has at least O (n1n2)as in [2] time complexity where n1 and n2 are the sizes of two DOM trees generated by a parser and the sizes of the trees are usually more than a thousand. Most existing methods as in [5][7] for template detection operate on a per website basis by analysing several web pages from the site and identifying content that repeats across many pages. The problem of extracting a template from the web documents conforming to a common template has been studied .In such systems we are assuming that web pages generated will have common template, the solutions for this problems are applicable only when all the documents will have a common structure as in [1][11]. However in real applications, it is not trivial to classify many crawled documents into homogeneous partitions in order to use these techniques.

A site-level template detection method has some limitations. First, site-level templates constitute only a small fraction of all templates on the web. Second, these methods are error prone when the number of pages analysed from a site is statistically insignificant.To overcome the limitation of the site level template detection, we are having a assumption problem of extracting the templates from a collection of heterogeneous web pages, which are generated from multiple templates as in [8][9][10] . In this problem, clustering of web documents in which all the documents related to a cluster contains common template paths, the correctness of template structure depends on the clustering.

II. RELATED WORK

To overcome the limitations of using DOM tree representation for similarity comparisons between the web pages, we are representing a web document and a template as a set of paths in a DOM tree. We are using XPATH technology of XML as in [12], where paths are sufficient to express tree structures and useful to be queried. By considering only paths, the overhead to measure the similarity between documents becomes small without significant loss of information and also it decreases the time complexity for comparing the web page similarity. To deal with the unknown number of templates and select good partitioning from all possible partitions of web documents, we employ Rissanen's Minimum Description Length (MDL) principle which represents the cluster with minimum number of bits.

We are using hierarchal clustering approach of the text based web document clustering. The text-based web document clustering approaches characterize each document according to its content in it. The basic idea is that if two documents contain many common words then it is likely that the two documents are very similar. In the proposed system we are using agglomerative hierarchal clustering algorithm for clustering the documents which produces a sequence of nested partitions.

For example, let us consider simple HTML documents and their paths in Figure. 1 and Table 1. Document d2 is represented as a set of paths {p1, p2, p3, p4, p5} and the template of both d1 and d2 is another set of paths {p1, p2, p3, p4}.Template path is a common path for group of documents. Support of the template path is defined as a number of documents which that particular path exists.

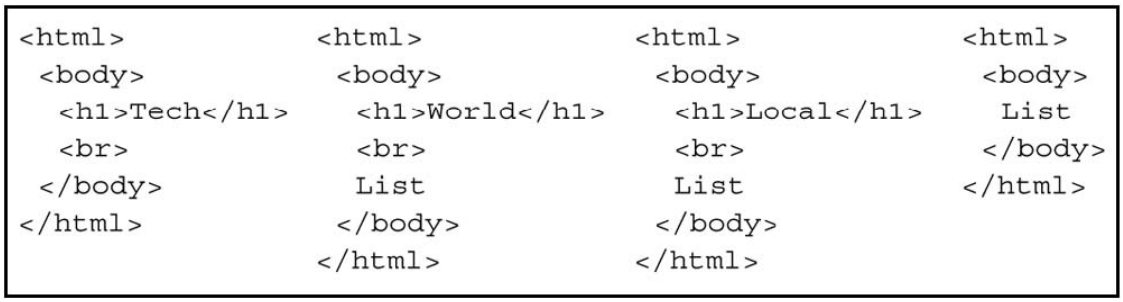

Figure.1. (a) Document d1. (b) Document d2.(c) Document d3. (d) Document d4.

TABLE 1
Paths of Tokens and Their Supports

ID	Path	Support
p_1	Document\⟨html⟩	4
p_2	Document\⟨html⟩\⟨body⟩	4
p_3	Document\⟨html⟩\⟨body⟩\⟨h1⟩	3
p_4	Document\⟨html⟩\⟨body⟩\⟨br⟩	3
p_5	Document\⟨html⟩\⟨body⟩\List	3
p_6	Document\⟨html⟩\⟨body⟩\⟨h1⟩\Tech	1
p_7	Document\⟨html⟩\⟨body⟩\⟨h1⟩\World	1
p_8	Document\⟨html⟩\⟨body⟩\⟨h1⟩\Local	1

However, it is not easy to select proper training data , since we do not have any knowledge about given data in advance. Moreover, it is hard to decide how many clusters are to be generated from given web documents in advance. To manage the unknown number of clusters we are using MDL principle proposed by Rissanen. And for handling large number of web documents we are using MinHash to calculate the MDL cost quickly.

III. PROPOSED APPROACH

A. *ESSENTIAL PATHS AND TEMPLATES*

The DOM defines a standard for accessing documents, like HTML and XML. The DOM presents a tree structure for an html document. The total document is a document node, every element of html is node, the texts in the HTML elements are considered as text nodes, every HTML attribute is an attribute node, and comments are comment nodes. we denote the path of the node by listing nodes from the root to the node for example, in the DOM tree of d2 in Fig. 2, the path of a node "list" is "Document\<html>\<body> \list."

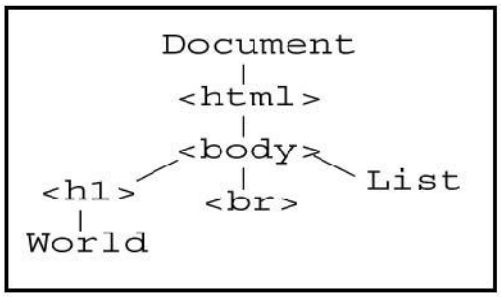

Figure 2. DOM tree of Document b in Figure 1

Given a web document collection D, we define a path set Pd as the set of all paths in Document set. For each document di, a minimum support threshold tdi will be assigned. If a path is contained by a document di and the support of the path is at least the given minimum support threshold , the path is called an essential path of that particular document di. We denote the set of essential paths of an HTML document di by E(di). For a web document set D with its path set PD, we use a |PD|×|D| matrix ME with 0/1 values to represent the documents with their essential paths. The value at a cell (I, j) in the matrix ME is 1 if a path pi is an essential path of a document dj. Otherwise, it is 0.

ME(paths X Documents) matrix for documents in Figure 1,where minimum threshold value is 3 for all the documents. The Essential path sets are E(d1)={p1,p2, p3,p4},E(d2)={p1, p2, p3, p4, p5}, E(d3)={p1, p2, p3, p4, p5}, and E(d4)={p1,p2}.

$$M_E = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

B. *Matrix Representation of Clustering*

Let us assume that we have N clusters for a web document set D. A cluster ci , Ti is a set of paths representing the template of ci and Di is a set of documents belonging to ci. To represent clustering information for Document set, we use a pair of matrices MT and MD, where MT represents the information of each cluster with its template paths and MD denotes the information of each cluster with its member documents. We will represent ME by the product of MT and MD. However, the product of MT and MD does not always become ME. Thus, we reconstruct ME by adding a difference matrix M_ with 0/1/-1 values to MT.MD, i.e., ME = MT.MD+ M_.

The Matrices for ME are given below

M_T M_Δ M_D

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

C. Minimum Description Length Principle

In order to identify unknown number of clusters and to select the good partition from all the clusters we use MDL principle by Rissan.MDL is the one-to-one correspondence between code length functions and probability distributions. For any probability distribution P, it is possible to construct a code C such that the length (in bits) of C(x) is equal to − log 2P(x); this code minimizes the expected code length. MDL is used to minimize the length of the model in bits and the length of the encoding of the data in bits. 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, Pr(1) and Pr(-1) are the probabilities of 1 s and -1 s in the matrix and Pr(0) is the probability of zeros. Then, the entropy H(X) of the random variable X is as follows

$$\sum_{x\in(1,0,-1)} -pr(X)\log_2 pr(X) \quad \text{and}$$

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

The MDL costs of MT and M_ is L(MT) and L(M_) respectively are calculated by the above formula. For MD, we use another method to calculate its MDL cost. The reason is that the random variable X in MD is not mutually independent, since we allow a document to be included in a single cluster (i.e., each column has only a single value of 1). Thus, we encode MD by |D| number of cluster IDs. Since the number of bits to represent a cluster ID is log2 |D|, the total number of bits to encode MD (i.e., LMD) becomes |D|. log2 |D|. Then, the MDL cost of a clustering model C is defined as the sum of the three matrices (i.e., L(C) = L(MT) + L(MD) + L(M_)). According to the MDL principle, for two clustering models C =(MT,MD) and C'=(M'T,M'D), we say that C is a better clustering than C' if L(C) is less than L(C').

By using the above formula we calculate the cost for each cluster generated by agglomerative clustering algorithm at different levels from bottom to up.At each stage the best cluster which have the less cost will be selected and given as input to next stage.

A clustering model C is denoted by two matrices MT and MD and the measure for it is MDL cost which is the sum of L(MT),L(MD),L(M Delta).When the pair of clusters are merged the cost may be increased or decreased. The Best pair is selected based on the cost for next level of processing until no reduction is possible. TEXT MDL deals with unknown number of clusters but the time complexity increases if the number of documents increases so we use MinHash approach to reduce the time complexity.

D. Cost estimation of MDL using MinHash

MinHash (the min-wise independent permutations locality sensitive hashing scheme) is a technique for quickly estimating how similar two sets are. It has also been applied in large-scale clustering problems, such as clustering documents by the similarity of their sets of words.

Jaccard similarity and minimum hash values

The Jaccard similarity coefficient of two sets A and B is defined to be

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

It is a number between 0 and 1; it is 0 when the two sets are disjoint, 1 when they are equal, and strictly between 0 and 1 otherwise. It is a commonly used indicator of the similarity between two sets: two sets are more similar when their Jaccard index is closer to 1, and more dissimilar when their Jaccard index is closer to 0.

Here in this approach we are not generating the template paths for each cluster, we are estimating the cost by using the signature of the cluster Ck which is calculated by taking the minimum signature value from the list of signatures for that set. After the clustering we need to post process to get the actual template paths for that

cluster. The Jacquard's coefficient can be estimated with the signatures of MinHash and clusters whose Jacquard's coefficient is maximum can be directly accessed in the signature space.

E. Road Runner For Data Extraction

Road runner is designed to make the data extraction process automatic which is based on the *ACME matching technique*, for *Align*, *Collapse*, *Match and Extract* – to infer a wrapper for a class of pages by analyzing similarities and differences among sample and the HTML pages of the class as in [1]. It compares given set of html pages and the source HTML codes, in order to find matching and mismatching parts between the documents and, based on this knowledge, progressively refines a common wrapper. The output wrapper generated by this technique is a grammar that can be parsed against the pages of the class to extract data items.

There are essentially two kinds of mismatches that can be generated during the parsing

(a) String mismatches mismatches that happen when different strings occur in corresponding positions of the wrapper and sample which are used to discover fields with different data base value.

(b) Tag mismatches, mismatches between different tags on the wrapper and the sample, or between one tag and one string. These are used to discover iterators and optional . The traditional Road Runner works only for common template pages. Now consider this system which takes the heterogeneous web pages from different websites as input and groups them into clusters based on similarity template structures. The resultant clusters are given as input to the Road Runner System which generates the Data extraction output for each cluster.

IV. CONCLUSION

The proposed method clusters the web pages based on the similarity of the data and also extracts the template for web pages in the cluster simultaneously. MDL principle used to manage with the unknown number of clusters and to select good partitioning from all possible partitions of documents, and MinHash technique is used to speed up the clustering process by making cluster cost calculation fast . Finally the resultant clusters are given to road runner system for data extraction process.

REFERENCES

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

[2] M.N. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, and K. Shim, "Xtract: A System for Extracting Document Type Descriptors from Xml Documents," Proc. ACM SIGMOD, 2000.

[3] Advances in web-age information management: 7th international conference... By Jeffrey Xu Yu, Masaru Kitsuregawa, Hong Va Leong

[4] .S. Dhillon, S. Mallela, and D.S. Modha, "Information-Theoretic Co-Clustering," Proc. ACM SIGKDD, 2003.

[5] H. Zhao, W. Meng, and C. Yu, "Automatic Extraction of Dynamic Record Sections from Search Engine Result Pages," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB), 2006.

[6] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards Automatic Data Extraction from Large Web Sites," Proc. 27th Int'l Conf. Very Large Data Bases (VLDB), 2001.

[7] V. Crescenzi, P. Merialdo, and P. Missier, "Clustering Web Pages Based on Their Structure," Data and Knowledge Eng., vol. 54, pp. 279- 299, 2005.

[8] D. Chakrabarti, R. Kumar, and K. Punera, "Page-Level Template Detection via Isotonic Smoothing," Proc. 16th Int'l Conf. World Wide Web (WWW), 2007.

[9] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "Fully Automatic Wrapper Generation for Search Engines," Proc. 14th Int'l Conf. World Wide Web (WWW), 2005.

[10] B. Long, Z. Zhang, and P.S. Yu, "Co-Clustering by Block Value Decomposition," Proc. ACM SIGKDD, 2005.

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

[12] XPath Specification, http://www.w3.org/TR/xpath, 2010.

[13] M.N. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, and K. Shim, "Xtract: A System for Extracting Document Type Descriptors from Xml Documents," Proc. ACM SIGMOD, 2000.

AUTHORS PROFILE

Mrs. Swapna goud is working as an Associate Professor, Department of Computer Science and Engineering in CVSR college of Engineering, Anurag Group of Institutions. She has 6 years of teaching experience.

Mr. Vishnu Murthy G received his B.E and M.Tech degrees in Computer Science and Engineering. He is having 15 years of teaching experience. He is presently pursuing his Ph.D. in JNTU, Hyderabad and is the Head of the Computer Science and Engineering Department, CVSR college of Engineering, Anurag Group of Institutions. He has organized and attended various workshops and conferences at National and International level. He has been the resource person for Institute of Electronic Governance and BITS off campus programs. He is the Life Member of ISTE, IEEE, ACM, CRSI & CSI. He had 5 publications in international journals and presented 2 papers in conferences.