A Methodology for Template Extraction from Heterogeneous Web Pages

Vidya Kadam
Student
Bharati Vidyapeeth College of Engineering & Research, Pune
vdkdm24@gmail.com

Prakash. R. Devale
Associate Professor
Bharati Vidyapeeth College of Engineering & Research, Pune

ABSTRACT

The World Wide Web is a vast and most useful collection of information. To achieve high productivity in publishing the web pages are automatically evaluated using common templates with contents. The templates are considered harmful because they compromise the relevance judgement of many web information retrieval and web mining methods such as clustering and classification and badly impact the performance and resources of tools that processes the web pages. Thus, the template detection techniques have received a lot of attention to improve the performance of search engines, clustering and classification of web documents. In this paper, we are presenting the approach to detect and extract the templates from heterogeneous web documents and cluster them into different group. The pages belong to each group should possess the same structure .This saves the time to find out best templates from a large number of web document and also saves the memory which is required to find out the best template structure.

.**General Terms**

Template Extraction, Template Clustering

.**Keywords**

MinHash, Minimum Description Length (MDL), parsing.

1. INTRODUCTION

World Wide Web (WWW) is a vast and mostly used to publish and access information on the Internet. To achieve high productivity of publishing, the web pages in many websites are automatically evaluated by using common templates with contents. For human beings, the templates provide users easy access to the contents guided by consistent structures even though the templates are not explicitly announced. For example fig1. There is the set of book pages from Amazon, the data in each book has the same schema that contains each page the title, list of authors, price of the book and so on. On both the pages, the title of the book appears in the beginning followed by the name of the author.

Fig 1. Book pages from Amazon

We cannot group the web documents by URL. In fig2. The pages look clearly different but their URLs are identical except the value of layout parameter. If we consider only URLs to group the pages then the pages from different cluster will be included in the same group.

Fig2. Different template of the same URL

An HTML document can be represented with the Document Object Model (DOM) tree. The web documents are considered as trees and many similarity measures for the trees have been investigated for clustering.

In this paper, we find the problem of detecting the templates from heterogeneous web documents and present novel algorithms called TEXT (Automatic Template Extraction). In this we propose to represent a web document and a template as a set of paths in a DOM tree. By using XML query language XPATH, paths are sufficient to express tree structures and useful to query. So, by considering only paths the overhead to measure the similarity between documents becomes small without significant loss of information.

The main goal is to manage an unknown number of templates and to improve the efficiency and scalability of template detection and extraction algorithms. To work with the unknown number of templates and select good partitioning from all possible partitions of web documents, we use Rissanen's Minimum Description Length (MDL) principle. To improve the efficiency and scalability to handle large number of web documents for clustering, we use MinHash.

2. RELATED WORK

The template extraction problem is categorized into two areas. The first area is the site-level template detection where the template is decided based on several pages from the same site.Crescenzi et al. [3] studied initially the data extraction problem in which the roadrunner extracts data template by comparing web page pairs. One page is considered as initial template, and the other page is compared with the template, which is updated when there are mismatches. Rajagopalan [2] introduced the template detection problem. Previously, only tags were considered to find templates but Arasu and Garcia-Molina [1] observed that any word can be a part of the template or contents. Vieira et al. [6] suggested an algorithm considering documents as trees but the operations on trees are usually too expensive to be applied to a large number of documents. Zhao et al. [8] concentrated on the problem of extracting result records from search engines. For XML documents, Garofalakis et al. [4] solved the problem of DTD extraction from multiple XML documents. While HTML documents are semistructured, XML documents are well structured, and all the tags are always a part of a template.

The other area is the page-level template detection where the template is computed within a single document.Lerman et al. [5] proposed systems to identify data records in a document and extract data items from them.Zhai and Liu [7] proposed an algorithm to extract a template using not only structural information, but also visual layout information.

3. ALGORITHM REQUIRED

Algorithm: Min-Hash
Input: Web Pages
1) GetBestPair(Clusters, Documents)
1.1) initial C={cluster1,cluster2….documentN} // here initially no of clusters= no of documents
1.2) for each pair clisterI,clusterJ of Clusters in C
1.3) min MDLCost=0
1.4) MDLCost=calculate MDLCost(clusterI, clusterJ)
If(min MDLCost> MDLCost)
min MDLCost==MDLCost;
Store pair(clusterI , clusterJ);
1.5) cluster pages which having less MDLCost than other pair
1.6) update Cluster Set C by merging best pair in one cluster.

```
Function MDLCost(clusterI , clusterJ )
{
Get all paths from each cluster template;
Calculate Mt, Md, H(x) using given formule.
Calculate L(c)= L(Mt )+L(Md)+L(Mdelta);
Return MDLCost
}
```

4. SYSTEM ARCHITECTURE

In this project we are providing different web pages as input to our system. Each web page has different or may be same document structure**.** After that we are parsing these web documents into an xml document using DOM model. After that we are finding out the paths or flow or document structure using its tag entry in the XML document. After that we are applying the MinHash algorithm to find out best pair pages from given input pages. Then we are classifying these best pair pages into different groups. We are recommending these groups to the user. This saves the time to find out best templates from large no of web document and also save the memory i.e. required to find out the best template structure.

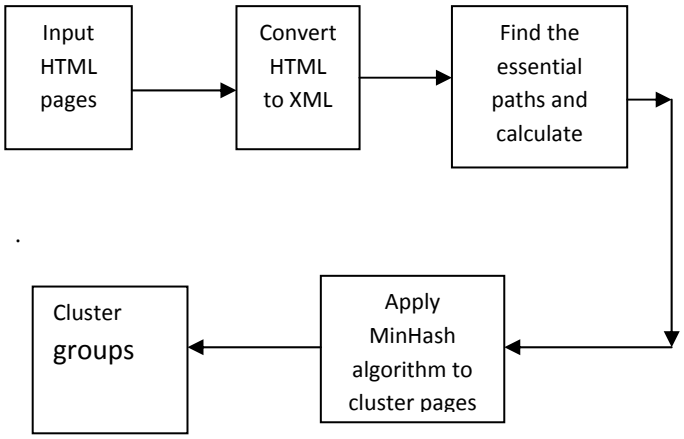

Fig 3: System architecture of Template Extraction from Heterogeneous Web Pages

We are clustering the page using two methods-:

1) Using the algorithm described above;

In first step we are converting html pages into xml pages . because html pages are semi structured and xml pages are well structured. After that we are finding the essential paths of each document and storing into the database. After storing the path we are finding the Mt and Md. Then we are calculating the MDL cost using the formulae described in algorithm. This step is recursively done to to calculate MDL cost of each pair . now if MDL cost of pair for ex document d1 and d2 is less than the d1 and d3 then that means the pair d1 and d2 can be cluster in one group. This is same for all the documents.

2) In this step we are treating each document as tree so tree have childs and depth. So we are calculating childs at each depth of document tree and comparing this depth and children with the others document depth and child. For example. If document d1 has depth 4 and d2 also have depth 4, then we are comparing each depth childs of one document with the others documents childs. Such as no. of childs at depth 2 of document d1 with no of childs at depth 2 of document d2. In this case we are finding out the mismatch between two document structure. If two document have less mismatch between their structure the these documents are clustered in one group.

5. CONCLUSION

We are presenting an approach for template detection from heterogeneous web documents .Each web page has different or may be same document structure. The best template from large number of web document saves the time of the user. We employ the MDL principle to select good partitioning from all possible partitions of documents and then we use a Min Hash algorithm to find out the best pair and to speed up the clustering process.

6. REFERENCES

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

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

[3] 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.

[4] 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.

[5] K. Lerman, L. Getoor, S. Minton, and C. Knoblock, "Using the Structure of Web Sites for Automatic Segmentation of Tables,"Proc. ACM SIGMOD, 2.

[6] K. Vieira, A.S. da Silva, N. Pinto, E.S. de Moura, J.M.B. Cavalcanti,and J. Freire, "A Fast and Robust Method for Web Page Template Detection and Removal," Proc. 15th ACM Int'l Conf. Information and Knowledge Management (CIKM), 20

[7] Y. Zhai and B. Liu, "Web Data Extraction Based on Partial Tree Alignment," Proc. 14th Int'l Conf. World Wide Web (WWW), 2005.

[8] 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.