International Journal of Scientific and Research Publications, Volume 4, Issue 8, August 2014 1
ISSN 2250-3153

Automatic Template Extraction from Heterogeneous Web Pages

Kunal Kumar Kundan, Professor Sonali Rangdale

Department of Information Technology, Siddhant, College of Engineering,Pune-33,India.

Abstract- In this paper, we will enlist the process of extracting template from heterogeneous Web Pages. Extracting structured information from semi-structured machine readable web pages automatically plays a major role these days, so some websites are using common templates with contents to populate the data for good productivity, Where WWW is the major resource for extracting the information. The problem here is for machines, the templates in the web pages are considered to be harmful since they degrade the performance of web applications due to irrelevant terms in the Template. As a result, the performance of the entire system degrades. Template Detection technique can be used to improve the performance of search engine as well as for classification of web documents. In this paper, we present algorithms to extract templates from a very large number of web pages that are getting generated from heterogeneous templates. Using the similarity of template structures in the document, we can cluster the web documents so that the template for each cluster will be extracted simultaneously.

Index Terms- Template extraction, clustering, minimum description length principle.

I. Introduction

Any HTML document can be represented by a Document Object Model (DOM) tree .web Pages are considered to be as trees and many similarity measures for trees have been investigated. However, clustering is very costly with tree related distance measure. For instance, tree-edit distance has O(n1 Log n2) time complexity[3],[4],[10],Where n1 and n2 represents the size of two DOM trees and the size of the trees are usually more than a thousand. Thus, clustering on sampled web page is used to practically handle a large number of web pages.

Since we are assuming that all documents are generated from only one template, solutions for this problem will be applicable to only when all documents are guaranteed to conform to a single template. However, in real time applications, it is not trivial to categorize massively crawled pages into homogeneous partitions in order to use this technique. The other part is the page-level template detection where the template is computed from a single document. Lerman[6] has proposed systems to identify different data records in a web page and extract that data items from those page. Zhai and Liu[8] has given an algorithm to extract a template based on structural information as well as on visual layout information.

A good template extraction method can relatively improve the performance issues of application. These days Fully automated wrapper generation which are used in search engine presents method which automatically produce wrappers that can be used to extract get the search result from dynamically generated which are returned by search engines .In real time application it is not good to traverse large number of pages and to differentiate them into homogeneous partitions. If we group web pages by using URLs, there can be different visual appearance of the documents . Hence we cannot group web pages by using these URLs.

In this paper we are proposing to represent a web page and a template as a set of different paths in any DOM tree. As validated by the most powerful XML query language XPATH, every path is sufficient to express the tree structure and is useful to be queried. When we consider only path, the overhead to measure the similarity between different web pages becomes small without much loss of information. In order to overcome this limitation of the above method, we have to develop such a method, which can extract templates from heterogeneous web document. But due to large number and different web pages , we have to manage unknown number of templates .This can be obtained by clustering web pages by selecting any good partition method. The effectiveness of extracted templates depends upon the quality of clustering.

II. Related work

There has been a number of recent work related to data retrieval. These can be categorize along different dimensions. Sources of data targeted example human vs. machine generated, degree of automation i.e complexity of data extracted example flat vs. nested. Section 1 briefly mentioned some of the related work. We are referring reader to a recent survey and tutorial for more data retrieval work. Here we try to focus on highlighting the differences between our work and ROADRUNNER[7] and IEPAD. IEPAD uses recurring patterns of closely occurring HTML tags to identify and extract records. This method is applicable for extracting data of a particular type, set of flat tuples from every page. Further, since not all recurring patterns contain useful information, IEPAD uses various heuristic method to categorize those that contain repeating data. This work is most closely related to the ROADRUNNER[7]. It uses a model of page creation that is very similar to ours depending upon the template. It starts with the first input page as initial template. Then, for each next page it checks if the page can be produced by the currently considered template. If it cannot then it changes its current template so that the changed template can generate all the pages seen so previously.

There are some limitations of the ROADRUNNER[7] approach:

ISSN 2250-3153

1. ROADRUNNER assumes that every HTML tag in the input data is produced by the template. This is important in ROADRUNNER to check if an input page can be produced by the recent template. This assumption is invalid for documents in many web pages since HTML tags can occur within specified data values. For example, a book review in any Web page could contain tags — the review could be in multiple paragraphs, in which some case it contains _j_ tags, or sometimes words in the review can be highlighted using _l_ tags. When the given input web pages contain such type of data values then ROADRUNNER will either fail to search any template, or generally give a wrong template.

2. It assumes that the grammar of the given template used to generate the documents is union-free. The authors[7] of ROADRUNNER themselves have specified in that their assumption that it does not hold for large number of collections of pages. Moreover, as the every experimental results suggest, ROADRUNNER may fail to give any output if there are too many disjunctions in the given input.

3. When it discovers that the current template does not produce any input document, it performs a complex heuristic search which involve "backtracking" for new template. This search is exponential with the size of the schema of the document. Therefore, it is not clear how ROADRUNNER will cope up to web document collections with a complex schema.

This paper[7] described an algorithm i.e EXALG, for extracting structured information from a heap of different web documents generated from a single common template. It first search the unknown template that produced the pages and uses the searched template to extract the information from the input document. It uses two concepts, equivalent class and different roles, to search the template. Our experiments on various collections of web documents, extracted from many well-known data sites, indicate that it is extremely useful in gettinh the data from the web documents.

III. PROPOSEDSYSTEM

Template extraction method consist of following steps :-

• HTML document and Document Object Model
• Essential paths of document
• Template of document
• Representation of clustering
• Minimum Description Length
• Clustering algorithm

For web pages, if we use HTML parser, we can parse the web page and create the DOM tree. From this DOM tree of web page we can find out its path[9] , there are only two types of path that is content path and the template path. Template paths will contain only structured data. It does not have actual contents of the data. From Document object model tree one can get the essential paths, which shows template. The system model can be shown in this figure.

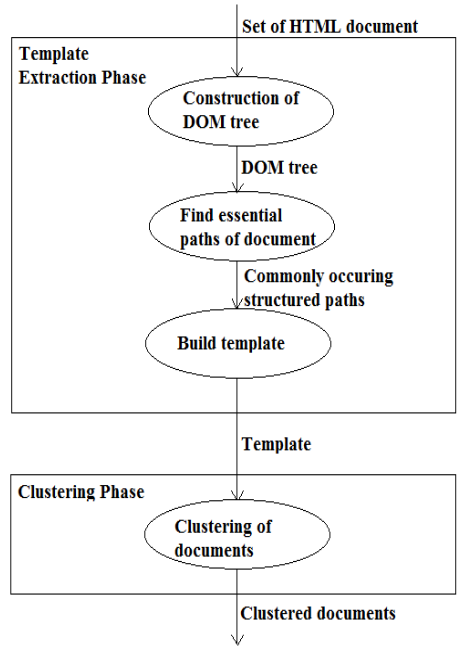

Figure 1: System model for template extraction

Essential paths of document

Define path set PW for document W,[7]. PW is set of all paths in W. Support of path is the number of pages in W, which contain the path. For all wi, there exists wi where twi is decided by taking mode of support values .If a path is contained by a document wi and support of path is at least the given twi then that path is considered as essential path of wi. Set of such essential paths are EP[9]. These essential paths are used in the extraction of template.

```
<html>
<head>
<title>DOM tuto</title>
</head>
<body>
<h1>DOM lesson</h1>
<p>Hello</p>
</body>
</html>
```

Figure 2: HTML Document

TEXT MDL ALGORITHM

The Minimum description length based compression principle produces a two types of code of the training data, one with the model portion of the code which is being used to compress and categorize test information[9]. We enlist a pseudo-code of the algorithm for creation model and exploration the conflicting requirement between reducing grammar size and reducing descriptive cost. We will show the results of a minimum description length model-based classification system for detecting network traffic anomaly.

ISSN 2250-3153

Procedure GetHashMDLCost(c_i, c_j, C)

Begin

1. $D_k := D_i \cup D_j$, $c_k := (\emptyset, D_k), C' : C - \{c_i, c_j\} U\{c_k\}$;
2. For each π_q in Π do {
3. $r(sigD_k[q]) := \min(r(sigD_i[q], r(sigD_j[q]))$;
4. If $r(sigD_i[q]) == r(sigD_j[q]$ then
5. $n(sigD_k[q]) := n(sigD_i[q] + n(sigD_j[q])$;
6. Else $n(sigD_k[q])$ is from the less one;
7. }
8. Calculate $\hat{\xi}(D_k, l)$ by Equation(5);
9. Compute $n(D_k, k)$ by Lemma 4;
10. Get $P_r(1)$ and $P_r(-1)$ in M_T and M_Δ by Lemma 3;
11. MDL := Approximate MDL cost of C' by Equation(1);
12. Return (MDL, c_k);

end

Figure3:GetMDLCost Algorithm

Agglomerative Clustering Algorithm :-

The algorithm forms clusters in bottom-up approach:

1. At the start, we will put each article in their own cluster.

2. Among all current clusters, we will choose the two clusters with the small distance than others.

3. We will replace these two clusters with a new created cluster, created by merging the two original cluster.

4. We will continue to do the above two steps until there is only one cluster left.

Thus, the clustering algorithm will result in a binary cluster tree in which single article cluster as its leaf nodes and all the articles are contained in the root node[13].

In this clustering approach, we will use a distance measure based on log. Consider articles A and B, the distance can be defined as

$$dA, B = LLA + LLB - LL(AUB)$$

A back-off trigram model [1] is created for every cluster, and interpolated with a trigram model which can be obtained from all articles for smoothing, to compensate different amounts of training data for every cluster. Then, the set of LMs that increases the log likelihood of the development data. Consider a cluster model set , LM= {LMi} the test set log likelihood can be received as an approximation to the mixture-of-various cluster model.

$$P(w|LM) = \sum_i P(LM_i) * P(w|LM_i)$$

$$\approx P(LM_{i^{\leftarrow}}) * P(w|LM_{i^{\leftarrow}})$$

$$\propto P(w|LM_{i^{\leftarrow}})$$

where $\quad i^{\leftarrow} = \underset{i}{\operatorname{argmax}} \; P(LM_i|A)$

$P(LMi)$and $PLMiA$ are the two prior and posterior cluster probabilities. In this training, A is the reference transcript containing one story from the Hub development information. While testing A is the 1-best hypothesis for this story, as described using the standard LM. $PwLM$ depends on the smoothing weights which is used to obtain $P(w|LMI)$ [1] which in turn compute which cluster a story has been assigned to, which in turn computes the best smoothing weight.

Procedure GetInitBestPair(c)

begin

1. Merge all clusters with the same signature of MinHash;
2. $MDL_{min} := \infty$;
3. **for each** c_i **in** C **do** {
4. $N :=$ clusters with the maximal Jaccard's coeff.with c_i;
   /*If the maximal Jaccard's coefficient is 0,N is $\emptyset$*/
5. **for each** c_j **in** N **do**{
6. $(MDL_{tmp}, c_k) :=$ GetHashMDLCost(c_i, c_j, C);
7. **if** $MDL_{tmp} < MDL_{min}$ **then** {
8. $MDL_{min} := MDL_{tmp}$;
9. $(c_i^B, c_j^B, c_k^B) := (c_i, c_j, c_k)$;
10. }
11. }
12. }
13. **return** (c_i^B, c_j^B, c_k^B);

End

Figure 4: GetInitBest Pair Algorithm

Procedure GetHashBestPair(c_k, C)
begin
1. (c_i^B, c_j^B) := the current best pair;
2. c_k^B := a cluster made by merging c_i^B and c_j^B;
3. MDL_{min} := the current best approximate MDL cost;
4. N := clusters with the maximal Jaccard's coeff.with c_k;
/*If the maximal Jaccard's coefficient is 0,N is ∅*/
5. **for each** c_l **in** N **do**{
6. (MDL_{tmp}, c_{tmp}) := GetHashMDLCost(c_k, c_l, C);
7. **if** $MDL_{tmp} < MDL_{min}$ **then** {
8. $MDL_{min} := MDL_{tmp}$;

Figure 5: GetHashBestPair Algorithm

Therefore, we will collectively try to optimize the cluster assignment in an iterative procedure. First of all, the posterior probabilities of the given cluster LMs reference[3] transcripts for a story were computed. After that, stories with the highest posterior probability of the same cluster LM are merged. The interpolation weight for the cluster LM is tuned by increasing the likelihood of the segments in the story cluster. This step is iterated until all the cluster assignments becomes stable.

IV. CONCLUSION

We introduced a new approach for the template extraction from heterogeneous web pages. We implemented the minimum description length principle for managing large number of clusters and to select good partitioning from all possible partitions of documents, MinHash technique in order to speed up the clustering process. Experiments with real time data, confirmed the effectiveness of our algorithms.

REFERENCES

[1] H. Zhao, W. M eng, Z. Wu, V. Raghavan, and C. Yu, Fully Automatic Wrapper Generation for Search Engines, Proc. 14th Intl ,Conf. World Wide Web (WWW), 2005.

[2] Document Object Model (dom) Level 1 Specification Version 1.0, http://www.w3.org/TR/REC-DOM-Level-1, 2010.

[3] A. Arasu and H. Garcia-M olina, "Extracting Structured Data from Web Pages," Proc. ACM SIGM OD, 2003.

[4] Crescenzi, G. M ecca, and P. M erialdo, "Roadrunner:TowardsAutomatic Data Extraction from Large Web ites," Proc. 27th Int'l Conf.Very Large Data Bases (VLDB), 2001

[5] K. Vieira, A.S. da Silva, N. Pinto, E.S. de M oura, J.M .B. Cavalcanti, and J. Freire, "A Fast and Robust M ethod forWeb Page Temp late Detection and Removal," Proc. 15th ACM Int'l Conf. Information M anagement (CIKM), 2006.

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

[7] J. Cho and U. Schonfeld, "Rankmass Crawler: A Crawler with High Personalized Pagerank Coverage Guarantee," Proc. Int'l Conf. Very Large Data Bases (VLDB), 2007.

[8] T.M. Cover and J.A.Thomas, Elements of Information Theory. Wiley Interscience,1991.

[9] 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.

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

[11] D. Gibson, K. Punera, and A. Tomkins, "The Volume and Evolution of Web Page Templates," Proc. 14th Int'l Conf. World Wide Web (WWW), 2005.

[12] K. Lerman, L. Getoor, S. Minton, and C. Knoblock, "Using the Structure of Web Sites for Automatic Segmentation of Tables," Proc. ACM SIGMOD, 2004.

[13] F. Pan, X. Zhang, and W. Wang, "Crd: Fast Co-Clustering on Large Data Sets Utilizing Sampling-Based Matrix Decomposition," Proc. ACM SIGMOD, 2008.

AUTHORS

First Author – Kunal Kumar Kundan, Department of Information Technology, Siddhant, College of Engineering,Pune-33,India., kunalkumarkundan@gmail.com
Second Author – Professor Sonali Rangdale, Department of Information Technology, Siddhant, College of Engineering,Pune-33,India., Sonali_rangdale@rediffmail.com