Web Page Sectioning Using Regex-based Template

Rupesh R. Mehta
Yahoo! R&D
Bangalore, India
rupeshm@yahoo-inc.com

Amit Madaan
Yahoo! R&D
Bangalore, India
amitm@yahoo-inc.com

ABSTRACT

This work aims to provide a novel, site-specific web page segmentation and section importance detection algorithm, which leverages structural, content, and visual information. The structural and content information is leveraged via *template*, a generalized regular expression learnt over set of pages. The *template* along with visual information results into high sectioning accuracy. The experimental results demonstrate the effectiveness of the approach.

Categories and Subject Descriptors: H.3.3 [Information Storage, Retrieval]: Information Extraction
General Terms: Algorithms, Design
Keywords: Site-specific segmentation, Site-specific noise elimination, Tree-based reg-ex

1. INTRODUCTION

Information Extraction (IE) and Information Retrieval (IR) over the Web is a challenging problem and is complicated further by the large volumes of data and multitude of content classes. Web pages are structured to include not only main content sections, like product information in a shopping domain, but also sections like navigation panels, copyright policy. Each section carries different importance and hence there is a need to segment a web page into a set of sections and determine their importance.

Majority of the existing approaches mainly consists of an optional page-level, rule-based web page segmentation step and section (or DOM node) importance detection step leveraging site specific information and/or page level spatial and content features. Our approach falls in the same category. However, unlike other approaches, our web page segmentation approach leverages structural information across pages via *template*, along with page level, rule-based information. This leads to robust segmentation quality. Unlike other approaches, our approach helps to detect less important sections local to a cluster of pages (i.e. part of a site), with high confidence, leveraging *template* learning.

2. OUR APPROACH

The proposed approach leverages site level information with the observation that- In a particular site, informative contents of web pages are often diverse in their actual content, and/or presentations (structure), whereas noisy content share common content, link, and presentation styles. Here, text, links and images embedded in tags in a web page is considered as 'content'.

Given a website the approach takes k web page samples from the site and learns the *template* over the DOM structure of those samples. It then learns site specific node, content importance using structural and content features repeating across pages. The approach matches each test page with the learnt *template*, segment the web page into set of sections, and assigns importance to each section, using *template* learning, and page level spatial and content features.

Template similar to [1], is a tree-based regular expression learnt over set of structures of pages within a site. Initial *template* is constructed based on structure of one page and then it is generalized over set of pages by adding set of operators, if the pages are structurally dissimilar. In addition to HTML tags, *template* generalization part deals with three operators, '*', '?', and '|' to denote multiplicity (denotes repetition of similar structure), optionality (denotes part of structure is optional), and disjunction (denote presence of one of the structures) in the structural data, respectively. In brief, *template* is a generalized tree-based regular expression over structure of pages seen till now. To illustrate, consider a *template* as- (A)∗B(C)?D(E|F), where A, B, C, D, E, and F represents set of DOM nodes and/or sub-tree in the structure. This *template* matches all pages having their HTML structure as ABCDE, AABCDE, ABDE, ABDF, ABCDF, etc. Further details on *template* can be found at [2].

Template helps to captures structural and content repetition across pages which is used to determine section importance. Also, *template* captures set of structurally similar items under a STAR (*) node helping segmentation process.

The approach is splitted in two phases as explained below:

2.1 Site Specific Learning

This phase involves learning of structural and content repetition across web pages as described below:

1. For each site, create and generalize *template* over k random sample web-pages.

2. During *template* generalization, compute or update value for each feature, if present, for each leaf *template* node, based on corresponding structure nodes. Set of features used are page support (PS) for each *template* node, PS for each image source feature, PS for link feature, and PS for each text feature mapping to *template* node. Here, PS for a feature/node is defined as number of pages containing the feature/node.

WWW2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

3. After generalizing *template* over k samples, compute the node support (ratio of PS of a node to sample size, k) and features noise confidence (ratio of PS of a feature to the PS of a node containing the feature) at each leaf *template* node. This step helps to capture noise local to cluster of pages.

4. Store noise confidence of content features at nodes whose node support is greater than some threshold (say, 20%).

2.2 Segmentation and Importance Detection

This phase involves segmenting the web page into set of sections and determining their importance, leveraging *template* and visual information as described below-

2.2.1 Template-based Scoring

Match each test page, with the learnt *template* and get mapping of each *template* node to corresponding set of structural nodes in a page. Transfer noise confidence score to leaf structure nodes based on the presence of content feature.

2.2.2 Web-page Segmentation

The segmentation process is described as follow:

1. Web page often contains list of same items like list of products or list of navigational links, whereas each item is represented by a set of HTML nodes. We treat such list as a section, as all items belonging to the list will have same importance. STAR ('*') *template* node in a *template* represents such list. Hence, all HTML nodes mapping to a STAR *template* node are treated as a part of a 'section'. Note that, the approach considers uppermost STAR node, if nesting of STAR nodes is found.

2. In above step, set of sections are obtained by looking at STAR nodes. We assume DOM tree with visual information (height and width of each DOM node) is available for the page and hence for the remaining page, following steps are performed in top-down fashion, to obtain set of sections.

- Cond1- Ratio of node's area to the web page area is greater than some threshold (say 15%). Area of a node is computed as node height multiplied by node width.
- Cond2- One of its children has sectioning tag like TABLE, DIV and satisfies Cond1.
- Cond3- One of its children has section separating tag like HR, FRAMESET.

3. If node satisfies (Cond1 AND Cond2), its children are processed recursively.

4. If node satisfies Cond3, Child DOM nodes between two section separators or between first node and first section separator or between last section separator and last node are treated as separate sections.

5. If none of the conditions satisfies, DOM node is marked as section.

6. Note that, all contiguous, inline, sibling rich text formatting nodes like B, EM, and I are considered as a section.

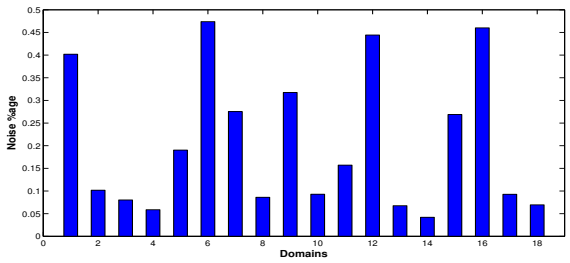

Figure 1: Average fraction of noise per domain

2.2.3 Section Importance Detection

Once segmentation process is over, each section is assigned importance score as: The noise confidence of each leaf structure node obtained in step (2.2.1) is aggregated at section level to determine the noise confidence of the section. The aggregation is weighted averaging of all noise confidence of leaf structure nodes based on their size. The section importance score is computed as (1 - section noise confidence). The importance value ranges between 0 to 1.

3. EXPERIMENTS

The approach is evaluated against 18 domains by randomly selecting 15 pages for learning and 65 pages for testing. Based on section importance, each section is classified into two categories- informative or noisy. If a section importance is less than some threshold (say, 25%), it is classified as noisy, otherwise informative. The evaluation of classified sections is done manually. Three person were presented with set of sections and its category and were asked to verify the sectioning quality and correctness of categorization. According to the evaluation, the approach could detect noisy sections with average of 91% precision and 82% recall. Also, it is found out that, the approach could form a section out of similar items (even with slight structure or visual differences), due to its *template* learning over set of pages.
The graph in Figure 1 depicts the domain-wise average of ratio of amount of noise detected in a page to the actual web page contents. The technique could detect an average of 20% of web page content as noise. It is also observed that, given sufficient and structurally slight varying training dataset, the approach was successful in detecting noise local to cluster of pages.

4. CONCLUSIONS

In this paper, we have solved the web page sectioning problem, leveraging site-specific information via *template*. A novel approach of *template* based web page segmentation and section importance detection is introduced. Preliminary experiments demonstrate the promise of the approach. The future scope involves further experimentation and incorporation of other features.

5. REFERENCES

[1] V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB '01: Proc of 27th Int'l Conf on VLDB*.
[2] V. G. V. Vydiswaran, R. R. Mehta, A. Madaan, and C. Tiwari. Tree-based template learning for high precision extraction. 2008.