EXTRACTING OBJECT-RELEVANT DATA FROM WEBSITES

Jianqiang Li and Yu Zhao
NEC Laboratories China,14F, Bldg.A, Innovation Plaza, Tsinghua Science Park, Beijing 100084, China

Keywords: Web data extraction, Web mining, Vertical web search, Web page clustering.

Abstract: This paper proposes a method to identify the object relevant information which is distributed across multiple web pages in a website. Many researches have been reported on page-level web data extraction. They assume that the input web pages contain the data records of interested objects. However, in many cases for data mining from a website, the group of web pages describing an object are sparsely distributed in the website. It makes the page-level solutions no longer applicable. This paper exploits the hierarchy model employed by the website builder for web page organization to solve the problem of website-level data extraction. A new resource, the Hierarchical Navigation Path (HNP), which can be discovered from the website structure, is introduced for object relevant web page filtering. The found web pages are clustered based on the URL and semantic hyperlink analysis, and then the entry page and the detailed profile pages of each object are identified. The empirical experiments show the effectiveness of the proposed approach.

1 INTRODUCTION

The research on web data extraction has been widely conducted (Laender et al, 2002; Kevin et al., 2004). Basically, the existing approaches are page-level solutions. They utilize inductive learning (Cohen et al., 2002; Kushmerick, 2000; Muslea et al., 2001) or deductive reasoning (Arasu and Garcia, 2003; Liu et al., 2003; Crescenzi et al., 2001) to find the template of target web pages for data extraction. Generally, the identified data records correspond to recognizable objects (or concepts) in the real world, such as products, projects, or persons. Based on the assumption that (1) the input web pages contain the data records of interested objects, and (2) all the information about one objects is contained in one web page, most of them can achieve reasonable accuracies in their evaluations.

However, a large amount of real-world objects are described in the current Web by multiple web pages (Zhu et al., 2007; Kevin et al., 2004). E.g., a product manufacturer might utilize multiple pages for describing each of its products. To extract the structural data of such objects, a new research, i.e., web data extraction across multiple web pages, is derived. Since the multiple pages relevant to an object are hyperlinked each other and serve as a constitutional part of the whole website, we call it website-level data extraction in this paper.

Different from page-level data extraction, the website-level data extraction assumes that (1) the group of pages describing an object are sparsely distributed in the website (only a small part of given pages contain relevant information of interested object), and (2) each relevant page contains only a piece of descriptions of the interested object. Since a website generally includes multi-types of objects (e.g., products and projects), the website-level solution needs to allow the user to specify the type of objects he/she is interested in. Therefore, the input of this task is also different from that of the page-level solutions, which comprises two parts: one is a whole collection of web pages of the website; and the other is the possible type information of the interested objects, e.g., the user might utilize the keyword “product” to represent that he/she wants to extract all the published products in a website.

These differences make the existing page-level solutions cannot be applied for the website-level data extraction. This paper describes our solution for this problem, which can serve as a complement technology to the page-level solution.

Intuitively, the solution of this problem can start with object identification (discovering the group of relevant pages of each object) and followed by applying the page-level techniques on the identified pages for structured data extraction. Due to space limitation and the fact that many page-level solutions have been reported, our focus is on the

Li J. and Zhao Y.
EXTRACTING OBJECT-RELEVANT DATA FROM WEBSITES.
DOI: 10.5220/0001823705970604
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

object identification problem, i.e., finding the entry page and other detailed profile pages for each object.

We examine how to utilize the hierarchical model adopted by website builder for the web page organization to help identify the interested objects in a website. Correspondingly, a two-stage strategy is adopted in the proposed approach: 1) object relevant web page filtering and 2) object identification based on hierarchical web page clustering. The first stage finds all the object relevant pages. Its basic idea is derived from following observations: Readers generally utilize the information appearing in multiple levels of hierarchical hyperlinks as a guide in website navigating; The anchor texts, URLs, page titles associated with the hyperlinks appeared in a reader's navigation path often give clear indication of the nature of the destination page. Then, the navigation paths are used to find all the object relevant pages in the website. The second stage first clusters the object relevant web pages into multiple groups, and then the objects are identified from each of the web page group. The underlying assumption is that a hyperlink referencing to an object generally selects the entry page of the object as the target. We validate our approach on the product extraction from 45 company websites in the IT domain. The result shows the effectiveness of the proposed approach.

2 RELATED WORK

Many research results have been reported to exploit the HTML DOM-trees of the web pages to discover the templates for the page-level data extraction. The earliest solutions (Arocena and Mendelzon, 1998; Hammer et al., 1997) mainly utilize the manual approach to acquire the knowledge of the template. To construct the wrapper through a semi-automatic way, supervise learning is adopted for wrapper induction (Cohen et al., 2002; Kushmerick, 2000; Muslea et al., 2001). For fully automatic web data extraction, several methods are proposed to discover the repeated patterns appeared inside a web page (Liu et al., 2003; Chang and Lui, 2003) or across multiple web pages (Wong and Lam, 2007; Crescenzi et al., 2001). Recently, the researches incorporating vision-based features for data extraction are reported (Cai et al., 2003; Zhu et al., 2006; Zhai and Liu, 2006).

Basically, the technologies discussed above can achieve reasonably accuracies. They all assume that target web page is generated automatically from a template. It causes the solutions can only be applied for the page-level data extraction.

This paper focuses on the problem of website-level data extraction. The most related work is the discovery of the logical domain (Li et al., 2001) or information unit (Tajima et al., 1998; Li et al., 2001). However, since their purposes are web search or search result organization, the solutions cannot be utilized directly for web data extraction. Also, the content-based solutions make the accuracy of resultant web page group not good enough.

For the website-level data extraction, our proposed approach differentiates the object relevant web page filtering as a distinct step. The time-consuming task for learning or page layout analysis on large scale web pages is avoided. Also the HNP based web page filtering can guarantee the high accuracy of the final object identification. Although the page layout analysis is involved in the following hierarchical web page clustering and page-level data extraction, since the target pages are only a small part of pages in the website, it is feasible for accurate object identification and data extraction.

3 WEBSITE-LEVEL DATA MINING

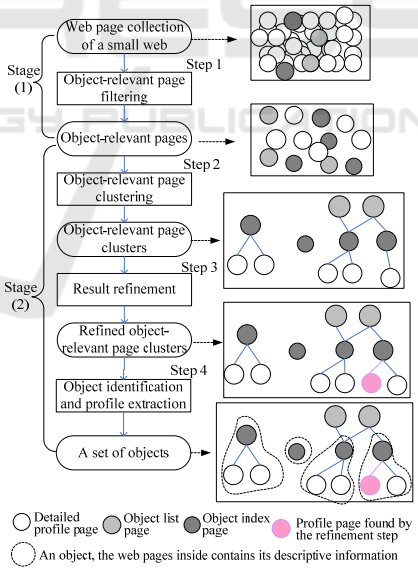

Figure 1: Procedures of object mining.

In this paper, we call the web page containing descriptions about the interested objects as the object relevant pages. They mainly refer to the object list pages (i.e., a navigation page including a list of hyperlinks pointing to multiple object entry pages) and profile pages (including certain profile information). For each object, its profile pages can

be classified into two types, i.e., the entry and detailed-profile pages. The entry page is the home page of the object. It contains a brief overview of the object, where the page reader can be guided to the detailed-profile pages. Each detailed-profile page describes one aspect of the object. The website builder generally uses one entry page and multiple detailed-profile pages for one object. Sometimes, there might be only one page includes both the overview and detailed information of one object.

Our proposed approach mainly comprises two stages, i.e., (1) object relevant web page filtering and (2) object identification based on hierarchical web page clustering. Concretely, it is embodied as four steps shown in Figure 1. The first step is web page filtering to obtain all the object-relevant pages from the page collection of corresponding website. In succession, the object-relevant web pages are clustered into multiple groups. The web pages in each group are linked together with certain hierarchical relations (HRs). In the third step, the result of above two steps is refined based on the layout based hyperlink analysis. The final step finds the entry and detailed-profile pages for each object.

3.1 Web Page Filtering

This step finds all the object relevant pages in a website. The traditional content based techniques (Baeza-Yates and Ribeiro-Neto, 1999) can be used here. However, the accuracy is not good, since the content of a page is generally not self-contained. We propose a HNP based approach for this task.

Formally, a hyperlink h is defined $h=<p_s, at, p_d>$, where p_s is the source page, at is the anchor text displayed for h, and p_t is the destination page of h. The constitutional hyperlinks of a HNP, which we call as Hierarchical hyperLinks (HLs), are different from those Reference hyperLinks (RLs) which convey the peer-to-peer recommendation, and also different from those Pure Navigational hyperLinks (PNLs) which provide just shortcut from a page to another page. HLs are utilized for web page organization and imply the HRs (e.g., whole-part or parent-child) between pages. Then the semantic of ancestor pages could be inherited to descendant pages along the HLs. A HNP is composed of multi-step HLs which constitute the assumed entry path to guide users' navigation from the root page to the destination page. It can afford meaningful indication on the content of its destination page.

3.1.1 HNP Construction

To extract HNPs from a website, first we need to realize the HL identification, i.e., to remove the PNLs as the noise information from the intra-site hyperlinks. The algorithm includes two parts: 1) syntactical URL analysis, and 2) semantic hyperlink analysis. The first part utilizes the directory information embedded in URLs to determine whether there is HR between the source and destination pages. The second part for semantic hyperlink analysis is adopted to find the PNLs that cannot be determined by the URL. The basic heuristic is originated from the observation that the destination pages of the hyperlinks from the same link collection (A link collection is a semantic block defined in Cai et al., 2003 that containing only a set of hyperlinks in a web page.) are siblings each other. Then, they should share the same parent and have different child pages. As shown in Figure 2, since the web pages in P_1 are outbound pages of a link collection (the rectangle in red line hosted by a web page), and their outbound hyperlinks $\{l_1, l_2, l_3, l_4\}$ share a same destination page in P_2, then $\{l_1, l_2, l_3, l_4\}$ are identified as PNLs. Similarly, $\{l_5, l_6, l_7, l_8\}$ are PNLs. Although the HL identification method is heuristic, the experiment has proved that it can discover most of the HLs in a website.

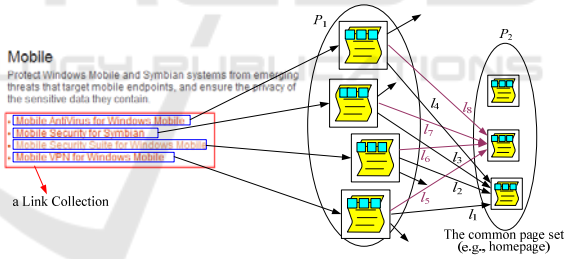

Figure 2: Semantic hyperlink analysis.

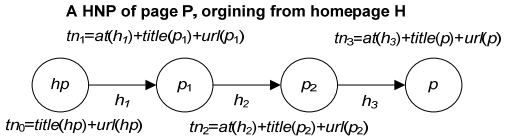

Figure 3: An example of HNP.

After the HLs are identified, the HNPs for each web page can then be generated by concatenating sequential HLs without a circular path, where the homepage serves as the root node of all these HNPs.

Since a HNP is utilized to discover the semantic meaning of its target page, the linguistic contents within HNP, i.e., the URLs, anchor texts, and page

titles along it, are collected. For a HNP <hp, h_1, h_2,…, h_n, p>, we represent $t(h_i)$ as the target page of link h_i, and then its linguistic contents as a text node list $\{tn_i\}$, i=0, 1, .., n, where tn_0=<$title(hp)$, $url(hp)$>, and tn_i=<$at(h_i)$, $title(t(h_i))$, $url(t(l_i))$>, i=1, 2, …, n. Figure 3 lists the text nodes for the example HNP.

3.1.2 HNP based Web Page Filtering

We formulate the object relevant web page filtering as a process of web page retrieval with given queries. More concretely, the query is modelled into two steps, i.e., path-query and content-query. For the path-query, HNPs serve as the intermedia between the query and the pages, i.e., whether a web page is relevant to the object is determined by its associated HNPs. The content-query is based on the result of path-query. It utilizes the page content to judge whether a web page is relevant or not.

The path-query is to retrieve object-relevant pages by querying the HNPs' text nodes with the object type names. For example, if a user wants to extract products from a company website, he/she might directly use the keywords such as "product" as the input query (the white keywords). If some text nodes of a page's HNPs contain such keywords, the page could be regarded as object-relevant. Optionally, in order to improve the performance of the path-query, a keywords blacklist including the names of other popular object types in the websites could be employed to filter out noise pages. That is to say, if the HNP of a page contains the keywords in the blacklist, the page could be regarded as a noise page. For example, if a page's HNPs include the keywords such as "news" or "projects" which is used popularly in the websites for objects other than products, this page might not be product-relevant.

A simple algorithm for HNP based filtering would treat each HNP as a document, therefore the similarity between the query and the HNP texts could be used to determine if the page is relevant. However, we found that the term frequency itself is not enough to determine if a web page is really relevant: white keywords (regarding to interested objects) and black keywords (regarding to uninterested objects) might coexist in different nodes of a HNP or in different HNPs of one page. The indication of a HNP on the content of the destination pages comes from not only its contained texts but also the position of the texts in the HNP. Correspondingly, we designed a weighting policy for object-relevant page filtering by use of HNPs: the position of the node containing the white/black-keywords is nearer to the destination page of corresponding HNP, more weight is given to the destination page indicating that it is object-relevant/irrelevant page.

Assume that HNP_p is the set of HNPs pointing to web page p. The HNP with maximum length in HNP_p contains N_p nodes (the length of a HNP is n means that it contains n text nodes). For a text node tn_i (i=0, 1, …, n-1) in a n-length HNP $hnp \in HNP_p$, the distance between tn_i and p along hnp is n-i-1. Since shorter is the distance, higher the weight of the node should be taken, we defined the weight of tn_i as $\boldsymbol{s}(tn_i)=N_p-(n-i-1)$. If tn_i contains any white keyword, let $\boldsymbol{w}(tn_i)=1$, otherwise, let $\boldsymbol{w}(tn_i)=0$. Similarly, if tn_i contains any black keywords, let $b(tn_i)=1$, otherwise, let $b(tn_i)=0$. Then, the decision function is defined as:

$$D(p) = \sum_{HNP_p} \sum_{tn_i} s(tn_i)[w(tn_i) - b(tn_i)]$$

Then, when $\boldsymbol{D}(p)>0$, p is considered as an object-relevant page, otherwise, it is a noise page.

Since the results of the path-query might contain not only object profile pages but also object list pages, the content-query is designed for filtering out the object list pages. It utilizes the unique ontological property names of the object type as keywords for further filtering. E.g.,, for describing a product, the unique property names include such as "function", "feature", and "price". The content-query is conducted on the results of the path-query. Since the property description appears only in the object entry or detailed-profile pages, the object list pages could be removed by the content-query. We need to mention here is that since the content-query depends on the pre-defined knowledge on the object, it is optional for the given solution in this paper.

3.2 Web Page Clustering

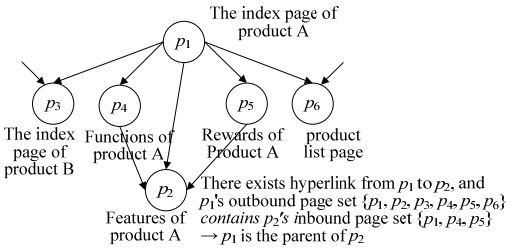

Figure 4: Example of HR identification.

After the object relevant web pages $P=\{p_1, p_2, …, p_n\}$ is obtained, this section describes a clustering technique to separate P into multiple groups. The basic idea is to adopt certain hyperlink analysis to

find the HRs between web pages, and then, naturally, each web page set linked together forms a cluster.

The above identified HLs can be utilized directly to construct the hierarchy of the object relevant pages. However, its underlying assumption depends too much on the existence of the link collection, and also the result's accuracy is not good enough for object identification. In addition to utilizing the URL for HR identification, we adopt a more general assumption here for discovering the HR between web pages: when the page author utilizes a hyperlink for reference to an object, he/she prefers to select the entry page of corresponding object as the destination of the hyperlink. So, for a detailed-profile page, its in-bound hyperlinks usually come from the object entry page or other sibling pages. Since all the detailed-profile pages are contained in the out-bound page set of the object entry page, the in-bound page set of the detailed-profile pages should be the subset of the out-bound page set of the object entry page. Thus, a basic rule for identifying the HR between two pages is: if there exists a hyperlink from page p_i to page p_j, and at the same time, p_i's outbound page set contains p_j's inbound page set, we regard p_i as an ancestor page of p_j. For simplicity, p_i itself is also included into its outbound page set virtually. Figure 4 is an example of using this rule.

```
//initialize L by hyperlink matrix H
L:=H;
//add virtual hyperlinks from pages to themselves to include page
itself into its outbound page set
l_ij:=1 for i=j;
//R represents the result of the previous iteration, R' represents the
result of the current iteration
R:=0; R':=D;
//iteration terminates when R becomes convergent
DO WHILE (R!=R')
{
    // initialize R' by directory matrix D
    R:=R'; R':=D;
    //adjust L by eliminating descendant-to-ancestor links and
    complementing ancestor-to-descendant (virtual) links by the
    resultant R of the previous iteration
    l_ij:=(~r_ji)&(r_ij|l_ij) for i ≠ j ;
    // if p_i's outbound page set contains p_j's inbound page set, p_i
    is an ancestor of p_j
    FOR EACH (p_i, p_j), i ≠ j
    {
        IF d_ji ≠ 1 AND l_ij==1 AND (FOR EACH
        1 ≤ k ≤ n HAVING l_ki ≥ l_jk )
          r_ij':=1;
    }
    //if two pages are ancestors of each other, remove their
    relation
    FOR EACH (p_i, p_j), i ≠ j
    {
        IF r_ij'==1 AND r_ji'==1
          r_ij':=0; r_ji':=0;
    }
    // derive the transitive closure of the hierarchical relation
    FOR EACH (p_i, p_j, p_k), i ≠ j ≠ k
    {
        IF r_ij'==1 AND r_jk'==1
          r_ik':=1;
    }
```

Figure 5: The algorithm for finding the HRs web pages.

Let $P=\{p_1, p_2, \ldots, p_n\}$ denotes the resultant page set of object-relevant web page filtering, where n is the amount of object-relevant pages. H is an $n\times n$ matrix. It represents the hyperlink relationships among pages of P, where the value of h_{ij} is 1 if there exists a hyperlink from p_i to p_j, and 0 otherwise. The $n\times n$ matrix D represents the HRships acquired directly from syntactic URL analysis, among pages of P, where d_{ij} equals to 1 if p_i is at a higher level than p_j in the directory structure implied in the URL, and 0 otherwise. Thus, an iterative algorithm for finding the HRs among the web pages in P is proposed, as shown in Figure 5, where P, H and D are the inputs, and the output, i.e., the resultant HRships among pages of P, are represented by $n\times n$ matrix R, where, $r_{ij}=1$ denotes that p_i is an ancestor of p_j, and $r_{ij}=0$ denotes there is no HR between them.

The essence of the algorithm is to construct R from H and D, where the syntactical URL analysis result is prior to that of semantic hyperlink relation. L is a virtual hyperlink matrix. Its initial value is set as H. For each cycle, L is updated by eliminating the descendant-to-ancestor hyperlink relation and complementing the ancestor-to-descendant hyperlink relation (i.e., virtual hyperlink). The termination condition of the iteration is that R converges.

It can be proved that, from the second iteration, the amount of page pairs with $l_{ij}=1$ decreases monotonically, and then the iteration is convergent.

After the HRs between pages in $P=\{p_1, p_2, \ldots, p_n\}$ are extracted, a graph of object-relevant pages could be generated by considering a page as a node and a HR between pages as an edge. Then the pages are naturally separated into multiple groups.

3.3 Result Refinement

Considering that both the web page filtering and clustering might raise some errors, which will be propagated to final object identification, we design a coordinated approach to refine the result of the two steps. The underlying idea is that the destination pages of the hyperlinks from the same link collection should be siblings each other. It means that if the destination pages of some hyperlinks in a link collection are judged as object relevant pages, the destination pages of other hyperlinks within the same link collection should also be object relevant.

Figure 6 shows an example. Within the hierarchical structure from clustering, the page p has the children p_1-p_4. p also has a link collection including the hyperlinks directed to not only p_1-p_4, but also two other pages p_5, p_6. The link collection could imply that p_5 and p_6 should be sibling of p_1-p_4.

Then, we coordinate the result by adding p_5 and p_6 as p's children.

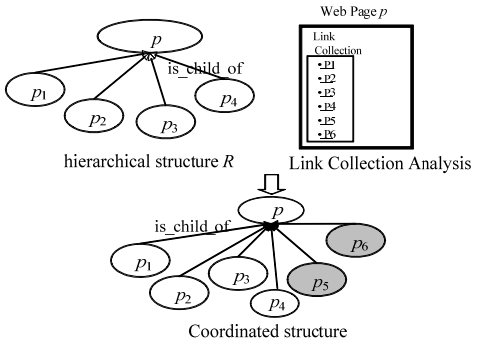

Figure 6: Coordinating R with Link Collections.

3.4 Object Identification

This section will identify each distinct object and its profile pages from those refined clusters. Since the entry page and the detailed-profile pages of each object are linked together with certain HRs, we can assume that all the profile pages of a specific object are completely contained within one cluster. Then, to find all the objects and their profile pages, we only need to consider each cluster one by one.

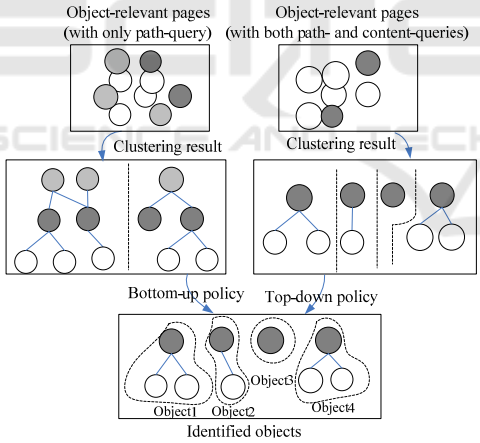

Figure 7: A schematic diagram on the object identification.

Depending on if the content-query is conducted, we design two policies for the object identification from each cluster:

Top-down policy: If the content-query has been done for web page filtering, i.e., the object list pages have been removed, we can consider the page without parent in a cluster as the entry page. Then, all its descendant pages are naturally identified as the detailed-profile pages of corresponding object.

Bottom-up policy: The bottom-up policy is employed for the case that the content-query is not conducted (i.e., the pages without parent in a cluster may refer to object list pages). The HRs between web pages determine that the details of the object are described in the lowest level pages of each cluster. Since the website builders generally use only one or two levels of pages for describing the objects, the bottom-up policy only investigates the pages at the two levels from the bottom of each cluster. Considering that an object entry page is the exclusively parent of other detailed-profile pages, and seldom two objects share one detailed-profile page, the bottom-up policy is embodied as two rules: 1) If a page at the lowest level has more than one parent, this page is an object entry page, which means its profile is described by only one page; 2) For a page at the second level from the bottom of the cluster, if all of its direct children have only one parent, this page is an object entry page and its children are the detailed-profile pages.

Figure 7 shows the application of two policies for object identification. The left part is the bottom-up policy applied on the clustering result of the object relevant web pages from only the path-query. The right part is the top-down policy conducted on the clustering result of the object relevant pages from both the path- and content-query. We can see that, no matter the content-query is conducted or not (which cause that the intermediate object relevant web page collection and clustering result are different), the final sets of the identified objects are the same. Our empirical observation shows that it is a typical scenario for website-level object identification, specifically the small granularity objects (e.g., products and people). However, for the object with large granularity, the profile pages may span more than two levels. It would make the results from the two policies different. In this case, content-query with the predefined knowledge of the general way that how these objects are described is preferred.

4 EVALUATION

4.1 Task and Data Set

Gathering the published product information in company websites can produce many value-added applications. A task to find out all the products and their descriptions in company websites is considered.

To set up a well-controlled experiment, a group of students are hired to help us select the dominate companies in the IT domain. We use three languages, i.e., English, Chinese, and Japanese, to verify if the

Table 1: Dataset and ground truth data.

	Websites	Total pages	Products (entry pages)	Product detailed profile pages
en	15	0.5M	1312	2753
cn	15	0.4M	1119	1692
jp	15	0.3M	632	855

approach is language dependent. For each language, 15 websites are considered. The representative companies include Symantec, McAfee, Cisco, Huawei, etc. Some global companies are picked in duplicate for different languages, e.g., the Symantec websites in three languages are selected respectively for English, Chinese and Japanese experiment. As shown in Table 1, we identify all the distinct products and their profile pages from these websites manually as ground truth data.

4.2 Evaluation Criteria

According to the convention of information retrieval, the Recall and Precision are adopted here.

For product-relevant web page filtering, the precision is defined as the proportion of actual product relevant pages among all web pages returned by the filtering approach. The recall is the proportion of the correctly identified product relevant web pages by the approach among all the product relevant web pages in the website.

For product identification, the qualities of the identified entry page and the detailed-profile pages are evaluated respectively. The recall/precision of the entry page identification reflects how well the products are discovered in the website; the recall/precision of the detailed-profile page identification reflects how well the detailed-profile pages are discovered for each specific product.

4.3 Experiment Results

We apply our approach for product extraction on these company websites. During the product-relevant page filtering stage, we conducted two experiments for the path-query, one by using whitelist keywords only, and another by using both whitelist and blacklist keywords. Then, the content-query is adopted to refine the web page filtering results from two kinds of path-query, respectively. Since our target is product mining, we define a product-specific keyword list for product-relevant page filtering. The pre-defined values are set as shown in Table 2. Subsequently, the steps of hierarchical web page clustering and results refinement are operated under two situations, with only path-query results and with both path- and content-queries results. Finally, the product identification is conducted.

Table 3 is the evaluation results of average recall/precision for the 45 test websites. Basically, the figures verify the usability of our proposed approach. The effective results of the product entry page and detailed profile page identification demonstrate indirectly that the performance of the given web page clustering method is promising.

Table 2: Parameter setting for experiments.

Filtering Phase: Path-query Keyword Whitelist	
en	product
cn	产品, 商品
jp	製品, 商品
Filtering Phase: Path-query Keyword Blacklist	
en	news, about us, bbs, forum, story, event, job, career, sitemap, service
cn	新闻,关于我们,论坛,市场活动,工作机会,招聘,网站地图,服务
jp	連絡, ニュース,会社情報,サイトマップ, フォーラム,伝言, 会社案内,サービス
Filtering Phase: Content-query Ontological Keywords	
en	function, specification, feature, benefit, advantage, performance
cn	功能,特性,特点,优点,优势,特色,亮点,规格,性能
jp	機能,特徴,特長,メリット,長所,仕様,スペック, 利点

Moreover, the figures in Table 3 show that the performance of our proposed algorithm is language independent. And the positive effect of considering blacklist in path-query is a bit, while considering content-query can increase accuracy notably.

Considering the incorrect results, one main reason is that some product entry pages are identified as product list pages, and then the actual product entry page is considered mistakenly as the detailed profile pages. For 11 such websites without product list page, the average recall/precision of product entry pages is 78.2/81.6. However, for the websites with product list pages, the average recall/precision of product entry pages is 80.8/86.3. Since the experiment of the later step is conducted based on the result of earlier steps, the errors are propagated from earlier steps to the later steps. So some errors in product identification is inherited from the errors happened in the steps of product relevant web page filtering and web page clustering.

Additionally, the identified HL for HNP extraction in web page filtering phase could also be used for finding HRs between object relevant pages, so we can use such HRs directly for clustering (called rough clustering) and product identification.

Table 3: Experiment results.

		Object relevant web page filtering				Product identification without refinement step (entry page / detailed profile pages)				Product identification with refinement step (entry page / detailed profile pages)			
		W	W+B	W+O	W+B+O	W	W+B	W+O	W+B+O	W	W+B	W+O	W+B+O
en	Rec. (%)	95.2	93.7	85.2	83.9	76.5/79.3	77.3/79.7	80.2/81.1	78.3/82.5	78.2/82.7	79.9/83.1	80.5/85.7	82.5/86.3
	Prec. (%)	79.4	81.2	82.5	89.9	72.6/80.9	75.2/81.3	81.1/82.7	83.1/83.2	74.2/83.2	75.9/85.7	81.9/87.0	87.3/88.5
cn	Rec. (%)	91.4	90.9	83.1	79.5	74.3/78.8	72.9/80.2	75.6/80.3	73.2/84.8	76.1/80.2	76.3/81.5	76.9/83.4	77.3/85.3
	Prec. (%)	75.1	78.3	80.0	88.8	71.1/77.5	75.4/77.9	79.9/81.2	81.0/81.9	75.9/78.0	75.8/79.1	80.4/81.6	82.1/83.6
jp	Rec. (%)	94.6	93.7	86.3	81.5	79.4/76.7	74.2/77.3	75.0/79.2	75.5/80.6	80.1/77.2	75.3/79.5	79.7/85.0	81.1/90.5
	Prec. (%)	80.2	81.5	81.3	89.2	72.9/79.1	75.8/79.7	82.5/82.3	84.8/83.5	73.2/80.3	77.0/82.1	85.7/82.8	86.8/84.0
Ave	Rec. (%)	93.7	92.8	84.9	81.6	76.7/78.3	74.8/79.1	76.9/80.2	75.7/82.6	78.1/80.0	77.2/81.4	79.0/84.7	80.3/87.4
	Prec. (%)	78.2	80.3	81.3	89.3	72.2/79.2	75.5/79.6	81.2/82.1	83.0/82.9	74.4/80.5	76.2/82.3	82.7/83.8	85.4/85.4

W: filtering with white wordlist of path query; **B**: filtering with black wordlist of path-query; **O**: filtering with ontological wordlist of content-query

The corresponding experiments are conducted to verify the effects of the adopted new web page clustering method in Section 3.2. Figure 8 is the comparison of the final product identification results with two web page clustering methods. We can see that the performance from the proposed clustering algorithm is enhanced significantly comparing with the results from the identified HLs. The impact of the refinement step can also be found in this figure, i.e., averagely, it improves performance about 3-8% with respect to both the recall and precision.

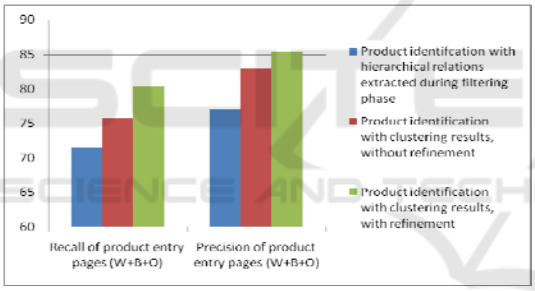

Figure 8: Comparison of product identification results.

5 CONCLUSIONS

Most existing solutions for web data extraction assume each given page includes several data records. It makes them not applicable for the problem of website-level data extraction, which assumes the relevant information of an object is distributed sparsely in the website. This paper proposes a novel approach to address this new problem. It exploits HLs for web page organization in websites as a novel resource for not only the object relevant web page finding but also the object centred web page clustering. The experiment results verify the usability of the proposed approach. And also some limitations exist in this method. The major limitation is that the object-relevant keywords need to be set manually. How to collect the keywords (semi-)automatically is our future work.

REFERENCES

Laender, A., da Silva, A., B. Ribeiro-Neto, and Teixeira., J., 2002. A Brief Survey of Web Data Extraction Tools. SIGMOD Record.

Arocena, G. O., and Mendelzon, A. O., 1998. WebOQL: Restructuring documents, databases, and webs. Proc. of ICDE.

Arasu, A. and Garcia-Molina, H., 2003. Extracting Structured Data from Web Pages. SIGMOD-03.

Liu, B., Grossman, R., and Zhai, Y., 2003. Mining data records in Web pages. In Proc. of the ACM SIGKDD.

Chang, C., Lui, S., 2001. IEPAD: Information extraction based on pattern discovery. Proc. of WWW

Cohen, W., Hurst, M., and Jensen, L. 2002. A flexible learning system for wrapping tables and lists in HTML documents. Proc. of WWW

Cai, D., Yu, S., Wen, Ji-Rong, and Ma, W.-Y., 2003. VIPS: a vision-based page segmentation algorithm. Microsoft Technical Report (MSR-TR-2003-79).

Hammer, J., Mchvoh, J., and Garcia-Molina, H., 1997. Semistructured data: The TSIMMIS experience. Proc. of the First East-European Symposium on Advances in Databases and Information Systems.

Davulcu, H., Vadrevu, S., Nagarajan, S., Gelgi, F., 2005 METEOR: metadata and instance extraction from object referral lists on the web. Proc. Of WWW.

Zhu, H., Raghavan,S., Vaithyanathan, S., 2007. Alexander Löser: Navigating the intranet with high precision. Proc. WWW.

Kao, H.-Y., Lin, S.-H., 2004. Mining web informative structures and content based on entropy analysis. IEEE Trans. on Knowledge and Data Engineering.

Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., Ma, W.-Y., 2006. Simultaneous record detection and attribute labeling in web data extraction. Proc. Of KDD

Park, J. Barbosa, D., 2007. Adaptive record extraction from web pages, Proc. WWW

Tajima, K. Mizuuchi, Y. Kitagawa, M., K. Tanaka., 1998. Cut as a querying unit for WWW, Netnews, and E-mail. In Proc. Of ACM Hypertext.

Kevin S. McCurley, A. T., 2004. Mining and Knowledge Discovery from the Web. ISPAN

Kushmerick, N., 2000. Wrapper induction: efficiency and expressiveness. Artificial Intelligence.

Muslea, I., Minton, S., Knoblock, C., 2001. Hirarchical wrapper induction for semi-structured information sources Autonomous Agents and Multi-Agent Sys.

Baeza-Yates, R., B. Ribeiro-Neto, 1999. Modern Information Retrieval. Addison-Wesley.

Wong, T.-L., Lam, W., 2007. Adapting Web information extraction knowledge via mining site-invariant and site-dependent features. ACM Trans. Internet Techn.

Crescenzi, V., Mecca, G. and P. Merialdo, 2001. Roadrunner: Towards Automatic Data Extraction from Large Web Sites, Proc. VLDB

Li, W. S., Ayan, N. F., H. Takano, H. Shimamura, 2001. Constructing multi-granular and topic-focused web site maps. Proc. Of WWW

Li, W., Candan, Vu, K. Q., Agrawal, D., 2001. Retrieving and Organizing Web Pages by Information Unit, Proc. Of WWW

Nie, Z., Ma, Y. J., Ma,W.-Y., 2001. Web Object Retrieval. Proc. of WWW.

Zhai, Y. H., Liu B. 2006. Structured data extraction from the Web based on partial tree alignment. IEEE Trans. on Knowledge and Data Engineering.