Simultaneous Product Attribute Name and Value Extraction with Adaptively Learnt Templates

Wei Tang, Yu Hong, Yan-hui Feng, Jian-min Yao, Qiao-ming Zhu
School of Computer Science & Technology, Soochow University, Suzhou, China
Email: hongy@suda.edu.cn

Abstract— **If we present the products as the attribute name and value pairs, it will improve the effectiveness of many applications. In this paper, we propose an adaptive template based method to simultaneously extract the product attribute name and value pair from Web pages. The titles of Web pages are used to assist the unsupervised template construction. And the template ranking strategy ensures the correct templates of every Web page are selected. Our approach contains four key steps: 1) construct domain attribute word bag by the titles of Web pages. 2) segment text nodes based on some default delimiters. 3) collect candidate attribute and value pairs 4) learn high-quality templates by a template ranking algorithm. The experimental corpus is collected from two domains: digital camera and mobile phone. Experiments show the precision of 94.68% and recall of 90.57% can be got by our method.**

Keywords- Web data mining; product attribute name and value pair; template construction.

I. INTRODUCTION

If we represent the products as a set of attribute name and value pairs, it will significantly improve the effectiveness of many applications that businesses use transactional data for, such as demand forecasting, product recommendations, assortment comparison across retailers and manufacturers, or product supplier selection.

The World Wide Web (WWW) contains a huge number of online shops which provides excellent resources for constructing the product database. Figure 1 shows a sample Web page, which presents the detailed information of a digital camera in the semi-structured manner. We regard this kind of Web page as ***detail page*** and the block containing product attributes as ***specification block***. For easy understanding, the product attribute names and values are presented contiguously. We regard such contiguous attribute name and value pair as ***NVP*** in this paper.

In this paper, we aim to extract as many NVPs as possible from C2C e-commerce sites to construct a commercial database. Compared to other types of sites, C2C e-commerce sites not only include more products but also have faster renewal speed. However different retailers tend to organize product attributes in different styles. Therefore how to achieve consecutive and efficient extraction is a great challenge.

Many existing information extraction methods are template-based. They can be categorized into two kinds: supervised learning approach and unsupervised approach. Supervised learning approaches often use several machine learning methods to infer templates, such as Wrap [14], WL2 [15], and [17]. However, manual labeling is labor intensive and time consuming and maintaining templates for different websites is a heavy burden. Furthermore, the templates have to be regenerated while the Websites change. Unsupervised approach, such as ExAlg [3], DEPTA [18], IEPAD [19] and DeLa [20], makes use of repetitive patterns in a Web page, or in HTML tags across multiple pages. However retailers organize the product attributes in the detail page in their private way, the similar layout format is rare in C2C e-commerce sites.

All the above works did not directly focus on the NVPs extraction, so it is difficult to be applied to our work. As far as we know, there are only few publications available on extracting NVPs from web pages. In [5], Bo and Chen use a co-training algorithm and a naïve Bayesian classifier to identify candidate NVPs in unlabeled pages. However it need training data. Wolfgang and Bernhard [6] use the ontology to extract NVPs from tabular data on Web pages. Two separate ontologies, table ontology and domain attribute ontology, are used to extract NVPs from a Web page. The domain ontology is hard to be constructed.

Besides, careful observation finds that a significant proportion of retailers organize the product attribute name and value pairs in the same text node. In mobile phone domain, the proportion is high up to 24.33%. However text nodes are treated as minimum extraction units in existing methods, so we can't get separated product attribute name and values in such kind of pages.

In this paper, we propose an unsupervised adaptive template based method to simultaneously extract NVPs from C2C e-commerce sites. We use all titles of the Web pages in a certain domain to construct domain-specific attribute word bag, which will be used to identify the candidate NVPs and provide a basis for the unsupervised template construction. Then the text nodes of every Web page are split into smaller segments through predefined delimiters. The smaller segment is referred as ***text fragment***. With the constructed domain attribute word bag, candidate NVPs in a page can be identified. Based on the

978-0-7695-4719-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CSSS.2012.503

candidate NVPS, the candidate templates are constructed. Finally by using a weighted word list and a dynamic threshold selection algorithm, the high-quality templates of every Web page are acquired from candidate templates. And the high-quality templates are used to extract other potential NVPs from the same page.

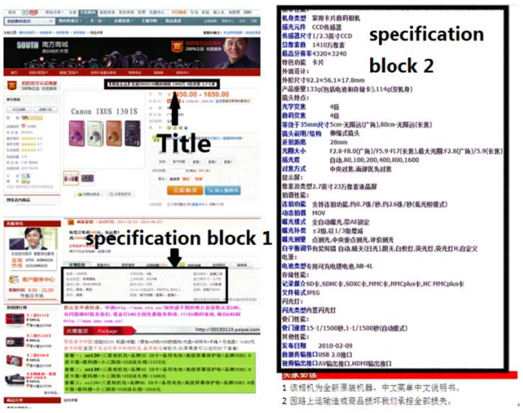


Figure 1. Example web page

The rest of the paper is organized as follows. Section 2 provides an overview of our method. And the works about preprocessing, candidate NVPs constructing, high-quality templates learning and template-base extracting are introduced in section 2. Then experiment result and analysis are showed in section 3. Finally section 4 concludes the paper.

II. ADAPTIVE TEMPLATE-BASED NVPS EXTRACTING

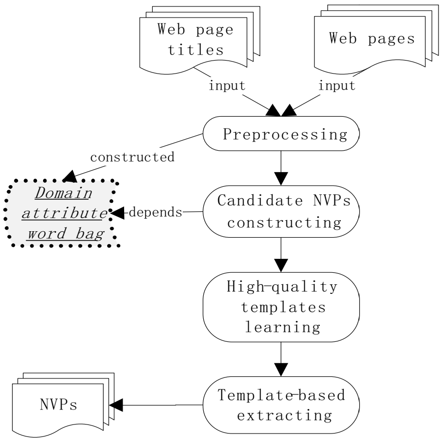

Web page titles
Web pages
input
input
Preprocessing
constructed
Domain attribute word bag
depends
Candidate NVPs constructing
High-quality templates learning
NVPs
Template-based extracting

Figure 2. The framework of our approach

In order to extract the NVPs from C2C e-commerce Websites, we proposed an unsupervised template-based method. As illustrated in Figure 2, our method contains four modules: preprocessing, candidate NVPs identification, high-quality templates learning and template-based extraction. The input contains two components: the first part is Web pages used to extract NVPs and the second is a set of Web page titles to construct the domain attribute word bag. The output of our system is the extracted NVPs.

In the section, we will present the details about the four steps in the proposed approach.

A. Preprocessing

Domain Attribute Word Package Constructing

We find retailers often design Web pages' titles by combining many attribute words related to the presented product. Take the Web page in Figure 1 for instance, its title is "*Genuine original Canon IXUS130 digital camera 14million 4times zoom 2.7-inch screen*", which many words are about product attribute. For instance, "Canon" refers to the brand name, "14 million" refers to the total pixel of the camera, and so on. And similar meaning or even the same phrases can also be found in specification block. They will greatly assist the candidate NVPs identification.

However, the attribute texts in a title are too rare, which will obviously affect the accuracy of identifying candidate NVPs. So we use many pages' titles in a domain to construct its attribute word bag.

A Chinese word segmentation system (ICTCLAS) is used to segment the titles into sequences of words. The title of example page in Figure 1 can be segmented into a consecutive word sequence, which is *"Genuine", "original", "Canon", "IXUS130", "digital camera", "14million", "4times", "zoom", "2.7-inch", "screen"*.

We can extract a large number of titles. All the titles are split into words, and form the domain-specific word bags. All the words are referred as ***attribute word***.

Text Node Segmenting

When a Web page is parsed into a DOM tree, the texts in Web page are all in DOM tree's leaf nodes, we refer them as *text nodes*.

According to the relative position of product attribute name and attribute value in a Web page, web pages are divided into two kinds:

1. The product attribute name and value, which constitutes a NVP, are organized in different but adjacent text nodes. For instance, "optical" and "4 times" can be organized following manner:

   <tr><td><span>optical</span></td><td>4 times</td></tr>

2. The product attribute name and value belong to the same text node. Just like the same NVP mentioned above, it also can be organized as the following:

   <br><span><span>optical ：4 times</span></span>

The existing methods usually assume that the web pages only meet one of the conditions. If we follow this train of thought and give up the treatment of the second case, it will greatly affect the recall rate. The further analysis shows that retailers often use some obvious separators, such as "：", to separate product feature name and value as in the second case, which provides probability for unifying the two cases.

We predefine a group of delimiters, which contains space, colon and symbol "-". These separators are matched in all the text nodes. And matched separators are replaced by the string "#segment#", just like the following example:

<br><span>optical ***#segment#*** 4times</span>

After this module, we get smaller processing units, which are referred as *text fragment* in the following steps.

B. Candidate NVPs Constructing

Potential Attribute Text Fragment Screening

We use following three clues to filter out noisy candidate attribute text: 1) the total number of attribute words that appear in the text fragment (***C***); 2) top first occurrence frequency (***F)*** of all attribute words in a text fragment; 3) the third clue is the length (***L)*** of the text fragment.

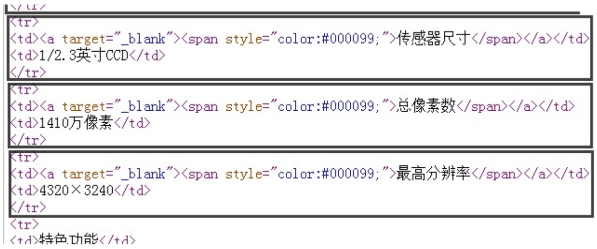
We define the high-quality text fragment as the text fragment that will help identifying the candidate NVPs. Base on the above three clues, we propose three hypotheses:

1. **Hypothesis I**: If a text fragment only describes a single attribute, it'll be regarded as high-quality attribute text fragment.
2. **Hypothesis II**: The product attribute words with lesser frequency are more likely to indicate a high-quality attribute text fragment.
3. **Hypothesis III**: The length of individual attribute name or value is smaller than common text.

In our experiment, we set ***C*** to be ***3***, ***F*** to be 2 and ***L*** to be 6.

Candidate NVPs Constructing

A NVP is composed of two adjacent text fragments. So these high-quality text fragments must be combined with adjacent text fragment to construct the candidate NVPs. But we can't determine these potential text fragments are about the attribute name or attribute value.



```
</tr>
<tr>
<td><a target="_blank"><span style="color:#000099;">传感器尺寸</span></a></td>
<td>1/2.3英寸CCD</td>
</tr>
<tr>
<td><a target="_blank"><span style="color:#000099;">总像素数</span></a></td>
<td>1410万像素</td>
</tr>
<tr>
<td><a target="_blank"><span style="color:#000099;">最高分辨率</span></a></td>
<td>4320×3240</td>
</tr>
<tr>
<td>特色功能</td>
```

Figure 3. Part of the HTML document of the example page

So we expend all the high-quality attribute text fragments in both directions. Every text fragment will be connected to the previous and latter text fragments. For example, if "*Total number of pixels*" in Figure 3 is the identified potential text fragments, it will be extended to two candidate NVPs. The extended NVPs will be defined as candidate NVPs. The bidirectional expansion also brings a lot of wrong candidate NVPs, in section 4.2.3, we will introduce the weighted word list to reduce the influence of these wrong NVPs.

C. High-quality Templates Learning

Candidate Template Constructing

We observe that the NVPs in a Web page display a repetitive pattern in a specification block. And the repetitive pattern is decided by the HTML code, so the corresponding HTML tags around the NVPs are also repetitive and can be used to construct the pattern.

When a candidate NVP is organized in different text nodes, the HTML tags around it are extracted. Take the NVP "*Total number of pixels*"-"*14.1 million pixels*" in Figure 3 as example, we construct a template below:

<tr><td><a><span>[tf]</span></a></td><td>[tf]</td></tr> (1)

"[tf]" refers to the text fragments about product feature name or value.

When a candidate NVP is organized in the same text node, the templates are still built by three parts. We still take the NVP above as example. If they are organized in the same text node, after segmentation of the text node, they may presented as the following in the corresponding HTML code: "<br><span><span>*Total number of pixels* #segment# *million pixels*</span></span>". The HTML tags around and the inserted string are used to construct the template. We get the template below:

<br><span><span>[tf] #segment# [tf]</span></span> (2)

We can find that the structure of the template (2) is consistent with template (1). Now we can find that the simple text node segmentation unify the template constructor.

We will count the type of the templates, and record the number of each type of each web page.

Weighted Word List

The bidirectional expansion of the high-quality text fragments brings many wrong NVPs, so many candidate templates are also incorrect. In order to obtain the needed template of a page, all the candidate templates will be ranked according to frequency. We introduce a weighted word list, in Table 1, to ensure that the high-quality templates can float to the top.

Table 1. Weighted word list

Category	Length	Weight	Times
United Word	m cm mm	kg g mg	s min h year

Many product attribute value are represented by number. If a text fragment is mainly composed of numbers and units, it has high probability of describing a product attribute value. Besides, the product attribute value is behind the product name in a NVP.

So if the second text fragment of a candidate NVP is mainly composed with numbers and units, its corresponding template will be weighted. The frequency of the corresponding template will be multiplied by a weighting coefficient.

In our experiment, the weighting coefficient we used is 15. Then all the templates will be sorted according the descending order of the frequency.

High-quality Templates learning

After the above steps, a set of templates corresponding to a page and their frequency are acquired. We need pick out the high-quality templates for a page.

Through observation, we find that the weighted templates are more likely to be high-quality templates. So we set a dynamic threshold which is adaptive to the frequencies of all the templates. The threshold is calculated by the formula below:

$$T = \frac{\sum_{i=1}^{n} Ap_i}{n}$$

In the equation, n indicates the number of the type of the template in a Web page. Ap_i indicates the frequency of the ith templates. Because the weighted templates' frequencies are much larger than the other templates', the threshold will be large enough to filter out the low frequency templates.

D. Template-based Mining

After high-quality patterns are selected, every two adjacent text fragments in the Web page will be treated as target NVP. Then we attempt to match each of the selected templates in the Web page. If the HTML tags around a target NVP were matched with one of the templates, the matched string pair will be kept.

The matching process is actually quite simple, since we transform the learnt patterns into standard regular expressions and then make use of existing regular expression matching tools (e.g., Microsoft .Net Framework) to extract NVPs.

III. EXPERIMENTS AND ANALYSIS

A. Experiment Corpus

Our experiments are in two domains, namely the mobile phone and digital camera domain. In each domain, the experimental corpus consists of three parts: the domain-specific Web page's titles, the product Web pages extracted from the C2C e-commerce sites, the product Web pages extracted from a technology portal and B2C e-commerce sites.

To construct the domain-specific attribute word bag, we extract 5000 titles of each domain from paipai.com and taobao.com, which are the largest two C2C e-commerce sites in china. We extract 4000 Web pages of each domain from the search result.

We also acquire Web pages from a technology portal, such as zol.com.cn and B2C e-commerce sites, such as 360buy.com and newegg.com. In digital camera domain and mobile phone domain, 500 Web pages are randomly selected from each Web site.

B. Baseline Methods

For comparison, we implemented two different baseline methods. One is the method proposed in [6], which use the table ontology and domain product attribute ontology, which contains 23 attributes, to extract the NVPs from Web pages. They just experiment in the digital camera domain.

The other baseline method assumes that the NVPs are organized in the same text node, so the text node segmentation module is removed. And the templates only contain HTML tag information.

C. Experiment Result and Analysis

Analysis of Corpus

We first analysis the corpus composed by the Web pages only from C2C e-commerce sites. These Web pages are organized by different retailer, therefore the organizational form are very different. The quantitative analysis of the corpus will help estimating the complexity, and provide support for our method.

We randomly extracted 300 Web pages from each domain. By the relative position of attribute name and attribute value of NVPs, they are divided into two categories. In the digital camera domain, the NVPs are organized in same text nodes in 44 Web pages. The ratio reached 14.67%. And in the mobile phone domain, the ratio is even high up to 24.33%. This phenomenon proves the necessity of the text node segmenting in our method.

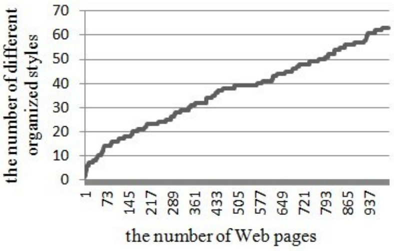


Figure 4. The relationship between the number of Web pages and the number of organized styles

To demonstrate the diversity of Web pages in C2C E-commerce site, we watch 1,000 pages under *paipai.com* site and 63 different styles are found. Figure 4 shows the linear relationship between the number of Web pages and the number of different templates. It demonstrates the difficulty of extracting templates manually. However, it is also difficult to get training corpus, when we try to extract NVPs with machine learning methods. This further shows the necessity of automatic extraction method.

Evaluation on the Web pages from C2C E-commerce Sites

We extract NVPs from the Web pages under C2C e-commerce sites by the mentioned two baselines and our method respectively. The final average evaluation result is showed in Table 2.

Table 2 Experimental results on extracting NVPs from the Web pages under C2C E-commerce sites

	Digital camera			**Mobile phone**		
	P	***R***	***F***	***P***	***R***	***F***
B1	0.8507	0.7809	0.8143	0.8269	0.6515	0.7288
B2	0.9720	0.8677	0.9169	0.9774	0.7569	0.8531
O	0.9567	0.9370	0.9467	0.9416	0.8744	0.9068

In the table *B1* and *B2* represent the first and second baseline method respectively. And *O* represents our method. This first baseline method heavily relies on the domain ontologies. As shown in Table 2, the first baseline method yields lowest precision and recall in both domains. Compared to the first baseline method, the second baseline method has great improvement.

We also observe that the recall value in digital camera domain is 11.08%, higher than that in mobile phone domain. The root cause is the corpus itself. Based on the second baseline method, our method adds text segmenting operation, and the character information is used to construct the template. Compared to the second baseline method, the recall in digital camera domain and mobile phone domain improved 2.98% and 5.37% respectively. However, the precision of the two domains declined. This is caused by some unreasonable text node segmentations. But the higher improvement in recall still brings us higher F score.

Evaluation on the Web pages from Technology Portal and C2C E-commerce Sites

Because the domain-specific attribute word bag is constructed only from Web pages' titles under C2C e-commerce sites, we design another experiment to test the feasibility of the attribute-specific word bag in technology portal and B2C e-commerce sites. The experimental result is shown in Table 3 above.

From Table 3, we can figure out that our method obtains better results. The reason is that these Web pages from the same site are consistent in the organization, so the weighted templates belong to the same type and they will have significantly higher frequency than the other templates finally. Besides, the NVPs in these sites are all organized in the same text nodes, so the recalls are greatly improved. The experiment result demonstrates that our method is highly versatile.

Table 3 Experimental result on the Web pages from 360buy.com, newegg.com and zol.com.cn

	Digital camera			**Mobile phone**		
	P	***R***	***F***	***P***	***R***	***F***
360buy	0.929	0.999	0.963	0.982	0.999	0.991
Newegg	0.884	0.998	0.937	0.920	0.999	0.990
Zol	0.986	0.999	0.992	0.999	0.927	0.962
average	0.933	0.999	0.936	0.967	0.975	0.970

IV. CONCLUSION AND FUTURE WORK

This paper proposes a new method to extract the NVPs from Web pages. In future, we intend to improve our approach in several directions, such as extracting NVPs from the Web pages in unstructured form. The titles can also be helpful, but new template constructing method should be proposed. We also want to apply our approach to other information extraction applications. The World Wide Web contains huge amount of resources. A general information extraction method will be of great significance.

[1] Liu L., Pu C., Han W. 2000. XWRAP: an XML-enabled wrapper construction system for Web information sources. In Proceedings of the 16th International Conference on Data Engineering (ICDE 2000). 1-22.

[2] Cohen W.W., Hurst M., Jensen, L.S. 2002. A flexible learning system for wrapping tables and lists in HTML documents. In Proceedings of the 11th international conference on World Wide Web (WWW 2002). 1-30.

[3] Zhai. Y.H., Liu. B. 2007. Extracting Web data using instance-based learning. In Proceedings of the 16th international conference on World Wide Web (WWW 2007), 113–132.

[4] Arasu A., Garcia-Molina H. 2003. Extracting structured data from Web pages. In Special Interest Group on Management Of Data (SIGMOD 2003).

[5] Zhai Y.H., Liu. B. 2005. Web data extraction based on partial tree alignment. In Proceedings of the 14th international conference on World Wide Web (WWW 2005).

[6] H. Alani, S. Kim, D.E. Millard. 2003. Automatic Ontology-Based Knowledge Extraction from Web Documents. In IEEE Intelligent Systems, Vol. 18, No. 1, pages 14–21, 2003.

[7] Wang, J., Lochovsky F.H.: Data extraction and label assignment for Web databases. In Proceedings of the 12th international conference on World Wide Web (WWW (2003). 187-196.

[8] Bo W., Cheng Xueqi, Wang Yu. 2009. Simultaneous product attribute name and value extraction from web pages. IEEE(2009), 295-298.

[9] 1 W. Holzinger, B. Krupl, and M. Herzog. 2006. Using Ontologies for Extracting Product Feature from Web pages. In Intenational Semantic Web Conferrence (ISWC 2006).