Mining Key Information of Web Pages

Chao Wang Jie Lu Guangquan Zhang

Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway, NSW 2007, Australia
{cwang, jielu, zhangg}@it.uts.edu.au

Abstract

Key information, in the form of distinctive menu items, navigation indicators provided by web site constructors, can classify the main contents of web pages and reflect certain taxonomy knowledge. Mining such information is significant as it can be used to improve web page classification and build up domain knowledge. This paper proposes a key information mining (KIM) method to extract this kind of information. This method contains two steps: a list of candidate key information is first extracted due to the criteria provided; an evaluation method based on entropy measure is then applied to discover key information. Experiment results show that the method is effective in discovering key information. With the discovered key information, classification of those web pages can be made easily and precisely.

Keywords: web mining, data mining, key information, classification, entropy.

1. Introduction

Web mining has become an essential research field due to the rapid growing data on the Web. Web content mining is one of web mining categories, mainly focusing on mining web pages, such as clustering or classifying page contents [1]. Whereas "noisy information" [8] in web pages, such as advertisements, copyright statements, and etc, is challenging common web content mining methods, some "key information" of web pages can help improve the data mining results if they can be utilized. Within a web site, if certain information embedded in web pages can categorize those web pages into several meaningful classes, we regard this kind of information as key information. For example, a distinctive menu item in a web page indicates the category of the main content in this page; a hierarchical navigation indicator shows the main topic of the page. Such menu items or navigation indicators can be considered as key information as they can categorize related web pages into different classes.

It is obvious that web page classification can be achieved precisely if key information exists and is fully utilized. Another advantage of using key information is to help catalogue integration [2]. Catalogue integration requires at least two catalogues that already have their categories. With key information be extracted and web pages from each web site be categorized, integration of them can then be proceeded. Moreover, as some kinds of key information usually reflect certain taxonomies, they can help build and populate ontologies that represent domain knowledge [3]. Thus, mining key information of web pages is significant for certain web related applications.

Wrappers [4] can be used to extract key information. However, as the styles of key information differ from site to site, manually creating wrapper for each site is time-consuming and error-prone. Wrapper induction [5] is an automatic way of creating wrappers based on a set of labeled examples. The main purpose of wrapper induction is to extract semi-structured data from web pages and to our knowledge there is no application of it to extract data like key information.

This paper proposes a new key information mining (KIM) method to ease the way of extracting key information embedded in the web pages. KIM does not require prior specific observations of certain web sites for creating certain wrappers, nor does it require any labeled examples. It generates a set of candidate key information and then employs entropy evaluation to select right information. With the KIM method, classification of web pages from certain web sites can be made easily and precisely. The rest of the paper will describe this KIM method in detail and then present related experimental results.

2. KIM method

The proposed KIM method consists of two main steps:

Step 1: Candidate key information list (CKIL) is extracted from each web page due to our given criteria; they are then merged to form a site candidate key information list (SCKIL);

Step 2: Entropy evaluation is applied to SCKIL so that real key information can be easily identified.

2.1. Step 1: Generation of candidate key information

It is effective for information extraction to treat a web page as a tree (DOM tree [6]) based on the HTML tags it contains. However, due to the flexible use of the HTML tags, it is not practical to rely only on the types of tags to extract the wanted information. Our observation shows that though key information (menus or navigation indicators) from different web sites has different styles, it does have something in common. It always contains concise words (around 2 or 3 words) and has consistent appearance within one web site. Fig. 1 gives examples of key information from two web sites, which illustrate this feature.

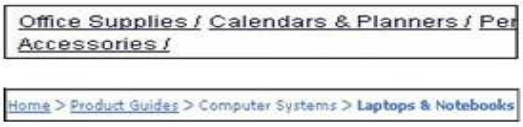

Fig. 1: Words in menus/navigation indicators

We first define some concepts so that descriptions of generating candidate key information (CKIL and SCKIL) can be made clearer.

Text node. We regard "CDATA" embedded in tags of a web page's DOM tree as a text node with the exception of those embedded in the tag <SCRIPT>, <STYLE> and <SELECT>, as texts in them are just functional codes, style or form data. An <IMG> tag is also treated as a text node if its attribute "alt" is set with values.

Word length. The word length of a text node is the number of words the node contains. Here words do not include separate characters (i.e. space, table character) or other meaningless characters. If the word length equals zero, the corresponding node will be ignored.

Menu subtree. A menu subtree is a sub tree of DOM tree of a web page. It is a subtree that contains only text nodes whose word length is less than α. The value of α can be adjusted. We commonly let α=5, as the word length of a menu item is always less then 5 based on our observation.

Menu item. Menu items are created from menu subtrees. A menu item *I* has two components, denoted by (*name*, *style*), where *name* is the text extracted from the text node of a menu subtree and *style* is the style pattern created by concatenating the tag information from the root of the menu subtree down to the leave of the text node. The tag information includes the name of the tag and its attribute lists.

Menu instance. A menu instance *MI* can be denoted as (*IList*, *Pages*), where *IList* is a list of menu items created from one specific menu subtree; and *Pages* is a set of web pages that have the existence of this menu subtree. An initially created *MI* from one page only has this page in its set of *Pages*. If a *MI*'s *Pages* contain a page *p*, then *p* is called to be *associated* with this *MI*.

Similarity of menu instances. Two menu instances MI_1 and MI_2 extracted from two different web pages are compared according to their first menu item, i.e., *IList[0]*. If either *IList[0].name* or *IList[0].style* is same, then they are *similar*; otherwise, they are not.Two similar menu instances may not be identical. MI_1 and MI_2 are *identical* (denoted by "=") only if $MI_1.IList$ is exactly same with $MI_2.IList$. If $MI_1 = MI_2$, then they can *merge* into one menu instance by adding up their associated web pages together.

Site menu. A site menu *M* consists of a set of similar menu instances extracted from different web pages of one web site, denoted by *MISet*. All identical menu instances in one site menu's *MISet* should merge together, remaining only similar instances.

Given the above definitions, CKIL can be denoted as menu instance list of a certain web page and SCKIL is corresponding to the site menu list.

It should be noted that the definition of the menu subtree is so loose that some other parts of the web pages can also become menu instances. However, as far as the menu containing key information can be extracted, the definition is acceptable, as we would perform an entropy evaluation later.

Fig. 2 is an illustrative example (adopted from http://www.pcmag.com) that can help understand these definitions. Both A and B are treated as menu subtree, as they only contain text nodes with word length less than 5. The first menu item of menu instance created from A has *name* "Home" and *style* " <a href=a1>". The two corresponding menu instances are similar because their first menu item is same. However, they are not identical due to the difference of their fourth menu item.

```
<div class="breadcrumb">
 <a href="a1">Home</a>
  &gt; 
 <a href="a2">
  Product Guides
 </a> &gt; 
 <a href="a3">
  computer Systems
 </a> &gt; 
 Laptops & Notebooks
</div>
```

A

```
<div class="breadcrumb">
 <a href="a1">Home</a>
  &gt; 
 <a href="a2">
  Product Guides
 </a> &gt; 
 <a href="a3">
  computer Systems
 </a> &gt; 
 Desktops
</div>
```

B

Fig. 2: Two illustrative menu subtrees

With the above definitions, it is easy to extract a list of menu instances from a given web page by treating it as a DOM tree and checking menu subtrees of it. After menu instance list is extracted from each web page, they can be merged into one site menu list if they are from one same web site. The merging process is controlled by the definition of similarity of menu instances and site menu. After the merge, a site menu list is generated for entropy evaluation.

2.2. Step 2: Entropy evaluation

Entropy evaluation has been demonstrated to be effective in finding informative content blocks [7] and eliminating noisy information [8]. Here we will show it is also effective in discovering key information.

There are three main categories of site menus. First, noisy information such as advertisements and copyright information would be treated as menus. However, such noisy information always constantly appears in the web pages across a web site. If they are treated as menus, these menus may only have one menu instance with many web pages associated. For example, there are 100 web pages from one web site *X*. A site menu containing an advertisement may have only one menu instance associated with 95 pages. Second, some parts of the main content can also be accidentally treated as menus. In this case the menu instances of these menus usually have little web pages associated. Continue the example of site *X*: an "accidental menu" may contain 2 instances associated with 2 and 3 web pages respectively out of 100 pages. The third category is menus containing real key information. Such menus always have several instances, each of which is associated with a number of web pages. For site *X*, a menu containing key information may have 4 instances associated with 10, 15, 20 and 30 web pages respectively. We regard a menu as a random variable with several values (instances) it can take. If the probability of taking each value is closer, the entropy of the random variable would be greater. Thus an entropy evaluation is suitable for distinguishing menus containing key information.

Suppose a menu *M* has *n* menu instances and the *i-th* instance has p_i web pages associated. The total number of web pages from the web site is *S*. Let $N = \sum_{i=1}^{n} p_i$, then $S \geq N$. The entropy of menu *M* can be calculated using the following formula by the definition of entropy [9]:

$$H(M) = \begin{cases} -\sum_{i=1}^{n} \frac{p_i}{S} \log \frac{p_i}{S} & S = N \\ -\sum_{i=1}^{n} \frac{p_i}{S} \log \frac{p_i}{S} - \frac{S-N}{S} \log \frac{S-N}{S} & S > N \end{cases}$$

Where *S>N* means that some web pages are not associated to any instances of menu *M*. In this case, we assume there is a "fictional instance" with *S-N* web pages associated.

It is better that the entropy of each menu be normalized to a fixed range of values so that they can be compared with each other. Given the property of the entropy: $0 \leq H(X) \leq \log r$, where *r* denotes the number of all possible values that random variable *X* can take [9], we calculate the normalized entropy for each menu using the following formula:

$$NH(M) = \begin{cases} \frac{H(M)}{\log n} & S = N \\ \frac{H(M)}{\log(n+1)} & S > N \end{cases}$$

Continue the above example of web site *X*. The normalized entropy of the menu containing advertisement is 0.29. For the menu containing accidental contents, its normalized entropy is 0.21. The menu containing real key information then has normalized entropy of 0.96. Thus menu with higher normalized entropy is more likely to contain key information that we expected.

In practise, the above evaluation method is sometimes fooled by some "false menus" who have only one instances associated with around half of the whole web pages from the analysed web site. These "false menus" are usually some fixed advertisements existing on a proportion of the web pages. To overcome this problem, we also calculate the normalized entropy (*NHx*) of the menus by excluding the "fiction instances" mentioned above. The final value used to evaluate the site menu is the averaged normalized entropy (*ANH*) of *NH* and *NHx*.

After *ANH* of all site menus is calculated given the above formulae, they are resorted according to this value. The new list of site menus is presented so that users can pick out the one containing key information by checking a few top ranked menus.

3. Experimental results

We collect web pages from five well known web sites: ABC news (http://abcnews.go.com), Yahoo news (http://news.yahoo.com), CNET (http://reviews.cnet.com), PC Magazine (http://www.pcmag.com), PC World (http://www.pcworld.com). The method used to collect web pages is a breadth first crawling within the given web site. Due to the large size of web pages one site can

have, the collected web pages are only a proportion of the whole pages from each web site.

We applied our method to web pages from each site to obtain an entropy sorted list of site menus. From this list, key information of the web pages can be discovered. Table 1 presents the experiment results. The column "Pages" denotes the number of web pages collected from the corresponding web sites. "List Length" means the length of candidate list containing extracted site menus. It is easy to see this list is quite long. "ANH" means the averaged normalized entropy value of our desired key information. "Rank" is its rank position in the candidate list. It is obvious that the rank position of the key information is quite up with a high averaged normalized entropy value. A little of "false" key information still has high values for some web sites, for instance, the CNET web site. This is because web pages from these web sites contain lots of small sentences or short phrases. These noises are all treated as potential key information, building up a very long site menu list (for CNET, 5837 long). Thus the effectiveness of the entropy evaluation is a little degraded. However, the results are still acceptable as the rank is still high compared with the long length of the candidate list.

Table 1: Experiment resutls for 5 web sites

Web site	Pages	List Length	ANH	Rank
ABC	671	484	0.929	1
Yahoo	492	558	0.924	2
CNET	786	5837	0.666	3
PCMag	700	1235	0.808	2
PCWorld	2360	847	0.877	1

Given the discovered key information, we are able to reorganize the web pages in a meaningful way. The web pages from web site PCMag, for example, can be classified as illustrated in Fig. 3. The numbers in the brackets tell how many pages are associated with the corresponding topics.

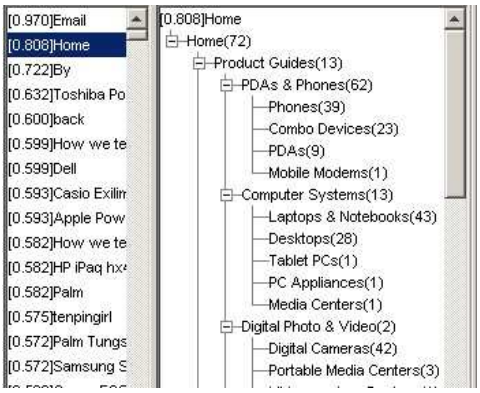

Fig. 3: Classification of web pages by key information

Besides the capability of finding a topic tree, the method can also find other key information provided by the web site. Still take the PCMag web site as an example. The 3rd ranked site menu contains the information of authors of the main contents. The web pages associated with this site menu are perfectly grouped by authors.

4. Conclusion

This paper points out the existence of key information in web pages and the significance of mining and using key information. It then proposes KIM, a two-phase method that can automatically mine key information from web pages. KIM first extracts a list of candidate key information and then uses an entropy evaluation to filter most of the noisy information in the list so that the key information can be discovered easily. An experiment shows that KIM is effective in mining key information, and key information can allow us to easily classify web pages in a meaningful way without applying sophisticated clustering or classification methods.

5. References

[1] R. Kosala, H. Blockeel, "Web Mining Research: A Survey", *SIGKDD Explorations Newsletter*, vol. 2 issue 1 pp. 1-15, 2000.

[2] R. Agrawal, R. Srikant, "On Integrating Catalogs". *Proc. of the WWW10*, pp. 603-612, 2001.

[3] H. Davulcu, et al, "OntoMiner: Bootstrapping and Populating Ontologies from Domain Specific Websites", *IEEE Intelligent Systems*, vol. 18 Issue 5, pp. 24-33, 2003.

[4] Y. Papakonstantinon, et al, "Object exchange across heterogeneous information sources", *Proc. 11th Int. Conf. of Data Engineering*, pp. 251-260, 1995.

[5] N. Kushmerick, "Wrapper induction: Efficiency and expressiveness", *Artificial Intelligence*, vol. 118 issue 1-2, pp. 15-68, 2000.

[6] http://www.w3.org/DOM/

[7] S. Lin, J. Ho, "Discovering informative content blocks from web documents", *Proc. of 8th SIGKDD*, pp. 588-593, 2002.

[8] L. Yi, B. Liu, X. Li, "Eliminating noisy information in web pages for data mining", *Proc. of 9th SIGKDD*, pp. 296-305, 2003.

[9] T. M. Cover, J. A. Thomas, "Elements of Information Theory", John Wiley & Sons, 1991.