A Web Information Extraction Method Based on Ontology

Wu Hengliang, Zhang Weiwei
School of Management Science and Engineering, Shandong Institute of Business and Technology, Shandong, Yantai, 264005, China
e-mail: hengliangwu@163.com, zwwc507@163.com

Abstract

Web information extraction is a very important and difficulty research subject which involves lots of fields, such as artificial intelligence, machinery learning, etc. As a modeling tool in describing the concept model of information systems at the semantic and knowledge level, ontology is widely used in many areas of computer science in recent years. A new method using ontology to extract valuable information from web documents was proposed in this paper. Firstly, according to the characteristics of the websites and web pages, the text content of web pages was extracted by locating the pages’ regional position. Secondly, on the basis of the traditional vector space model as well as the domain ontology, the concept vectors were generated according to the weightings of the concept vectors combining with the level structure feature of the ontology. Thus, the instances of ontology knowledge base were created semi-automatically, and the text of non-structured web page was turned into semantic structured information which can be understood by the machine.

Keywords: Web Extraction, Text Mining, Ontology

1. Introduction

With the development of Internet and web technology, WWW has been developing as a giant dynamic information serve network which includes many information sources and websites. It provides valuable information for the users. People can easily access to all kinds of information using Internet. However, the web information is unstructured and has no clear semantics, which makes the users find it difficult to get the useful information. The current search engine is keyword-based match, which makes the users cannot find the information fast and exactly.

In order to use the information sources on the web effectively, people put forward the concept of web information extraction and make the web become the knowledge base used by people in any time [5]. It offers the completely transparent, intelligent and uniform information access interface for users. Nowadays, more and more attentions are paid on the web information extraction technology based on ontology [2, 3, 10].

2. Ontology

Ontology is a term of philosophy, which is primarily used to describe the essence of the things [1]. Neches et al first proposed ontology in 1991, ontology was defined as the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary [5]. And in 1993, Gruber put forward a most popular concept that ontology is an explicit specification of a conceptualization [1]. Studer et al defined the concept from four aspects including conceptualization, formal, explicit, share etc [4]. An abstract model of a phenomenon termed “conceptualization”, a precise mathematical description hints the word “formal”, the precision of concepts and their relationships clearly defined are expressed by the term “explicit” and the existence of an agreement between ontology users is hinted by the term “shared”.

To date there is still no uniform classification standard about ontology [6]. We can classify ontology by different points of view, such as formalization degree, aims and objects described. For example, according to the objects described, ontology can be classified as domain ontology, ontology-based question solution, representation ontology and so forth. And an ontology describing objects in a certain field is called domain ontology, such as medicine, geography, finance etc.

With the development of the artificial intelligence, ontology is redefined. In the web information integration, ontology is often used to standardize the concepts and terms in a certain field or several
fields, providing a uniform concept and term standardization for the heterogeneous web data sources [3]. Therefore, it decreases the semantic conflict due to the web data sources adopting different names. In some cases the problem of semantics heterogeneousness is solved. Furthermore, it improves the accuracy of the system and provides more valuable information for the users [6-8].

3. Realization of web information extraction based on ontology and DOM tree

The process of web information extraction based on ontology is the mapping from web data to terms in a domain specific ontology. Firstly, this study tries to fetch texts from the web pages by locating information nodes in the corresponding DOM trees established according to the structure characteristics of web pages; extracts semantic information from the fetched texts and generates many concept characteristic vectors based on ontology base; then computes the weightings for these characteristic vectors and generates instances of ontology knowledge semi-automatically; transfers web data into semantic information comprehensible for applications, thus the task of web data extraction is finally finished.

3.1. Construction of domain ontology

Domain ontology refers to the abstraction and description to the concepts and knowledge in a field, which can exist and be used independently. It is still a hard task to construct a domain specific ontology at present, which needs industry experts to sum up information and knowledge in a field. The building methods mainly include: Uschold&King’s skeleton method, Methontology method, Gruninger&Fox’s TOVE method, Berneras method etc [7, 8]. Among them, the skeleton method is concluded by Uschold and King during the development of Enterprise Ontology, mainly includes the following steps, as shown in Figure 1.

3.2. Construction of websites' structure model based on DOM tree

On the basis of the analysis and researches to numerous websites, we divide web pages into two categories: directory web pages and content web pages. Among them, directory pages only provide a series of navigation links and don’t provide text content, that is, it’s just responsible for providing links for content pages. On the contrary, content pages provide real information that viewers want to obtain, which describe one or more topics by using lots of texts or pictures related.

At the same time, as most websites are based on hierarchical structure rather than network structure, we can map a website containing many relative pages into a digraph \( G = (N, E) \), as shown in Figure 2, in which each page can be considered as a node \( N_i (N_i \in N, 1 < i < k) \) of the digraph, \( E_j (E_j \in E, 1 < j < m) \) can be looked as an arc corresponding to the link between nodes, and each node has a URL containing the directory path of a node in the websites.
To conduct web information extraction, we first need to learn some basic knowledge about HTML which is the abbreviation of Hypertext Markup Language. Currently web information are mostly released in HTML format in which web documents are divided into a lot of components via some descriptive tags, such as paragraph, title and form, etc. No matter how complex a web page code is, its structure always includes those specific tags, and most are paired representing different meanings.

DOM (Document Object Model) is a set of standard interface specifications formulated by the W3C, providing access to documents data and structure. A parser is used to read a web document and construct an object model called DOM according to the tags’ relationship in the document, as depicted in Figure 3. This model describes a tree structure of the document, and each element in the document can be visited and processed through traversal to the DOM tree.

Because each HTML page corresponds to a DOM tree, the analysis methods based on DOM tree can learn the structure and relationship between various HTML tags in the page easily, thus web information extraction can make full use of these structure information. At the same time, we can easily get information about these tags with lots of DOM interface methods.

3.3. Extracting from web pages

Two different web information extraction methods are proposed in this study according to two different types of web pages. To directory web page we need to analyze its structure, locate its column area, and extract its relevant column headlines and links. Different from of the former, extracting from content page needs to get rid of page noises, locate its content text area, and extract its body text.

During the process of page text extraction we use two open source software: HTML Parser and JTidy. Among them, HTML Parser is a HTML parser containing a lot of tags and filter classes used to parse web pages. JTidy provides a grammar checker and a tag compensator with which all kinds of non-standard writing HTML documents can be repaired in order to make them parsed properly. We describe the page text extraction process in the following sections, as shown in Figure 4.
1) Web page capturing

In order to capture web pages automatically we develop a web page accessing sub-module in charge of sending requests to web server and receiving return messages from the server. This module can get HTML web pages from network at specified time intervals by using the URL and the function called ConnectionManager( ) in the HTML Parser, and put them into the local computer's memory buffers at last.

2) Web page pretreatment

On one hand, because a few HTML documents are not written abiding by the standard, such as short of end tag, an open source tool called JTidy is used to scan and repair each web page with some defects in this study. On the other hand, there are a number of web page noises that have nothing to do with the pages, such as page header, page footer, advertisements, images, scripts, etc. According to the HTML tags we design a web page pretreatment module for this study in order to remove and filter all of those noises.

After these two above page pretreatments, this study begins to parse HTML pages and create DOM trees by using the Parser( ) function provided by the HTML Parser. Web page pretreatment algorithm is as follows:

**Algorithm: Web page pretreatment**

**Input:** web page content

**Output:** DOM trees pretreated

**Begin**

JTidy.Process(Content);// repair web pages

myParser=Parser.createParser(content,"GB2312");// construct HTML parser

myParser=Parser.createParser(content,“GB2312”);// parse web pages into DOM trees

OrFilter Filter=new OrFilter(new NotFilter(new TagNameFilter("script")), new NotFilter(new TagNameFilter("style")));// construct HTML filter

NodeList nodelist=parser.parse(Filter);// filter useless tags

**End**

3) Extracting from the web pages

For one thing, we need to extract information from directory web pages which sum up lots of important information about the page text. This paper uses a match method between the terms in a domain specific ontology and the nodes in a DOM tree to obtain a minimum tree related to the column topics.

For another, we start to extract data from content web pages. According to the titles and URL extracted from the directory web pages, we can find the precise position of page text quickly. Based on the analysis to the content web pages’ HTML code, we find that the titles always appear in front of the page text, the page text appears between <P> and </P>, and they always exist in the same area. On basis of such features, an algorithm about page text extraction is designed for this study, details as follows:
3.4. Semantic processing to web information

We hope to indicate the major idea from the extracted page text $T$ combining statistics method and semantic analysis method based on ontology. On account of the defects of traditional vector space model in semantic processing, we put forward a concept vector space model that can express the major idea of pages effectively by introducing terms of ontology and converting document vectors into concept vectors composed of terms of ontology. The following gives the process of concept vectors based on ontology, as shown in Figure 5.

1) Chinese words segmentation

Word is the basic unit of Chinese, and there is no obvious separator between words, therefore, we should try to divide $T$ into a number of independent Chinese words $T_i$ ($T_i \in T, 1 < i < n$) above all. In this study we use a mixed method based on thesaurus and statistics to segment $T$ into words.

2) Ontology characteristic terms matching

If a certain $T_i$ can correspond to $O_j$ that is a certain concept node in ontology base, we can take $O_j$ as a characteristic term of the page text. Furthermore, because one concept sometimes may have many different expression forms, therefore, something must be done to remove repeated concepts in order to reduce the complexity of computing weightings of concept vectors.

3) Computing weightings of the concept vectors

After the match, concept characteristic vectors have been established, thus we get a concept set defined as $O = \{C_1, C_2, ..., C_m\}$. Then, counting the emergence times of each concept vector in $T$, and computing the weighting of each vector, whose aim is to show the major idea of the page correctly and lay the foundation for the relevance sorting of vectors later.

This paper uses an amended TF-IDF method to compute each vector’s weighting, as shown in the following formula:

$$W_i = \lambda \times tf_i(T) \times \log\left(\frac{N}{n_i}\right)$$

(1)

Here, $tf_i(T)$ refers to the frequency of $t_i$ in $T$, $n_i$ is the document number containing $t_i$, $N$ is the number of documents, $\lambda$ is the weighting coefficient. The bigger the weighting of a certain
concept, the more relevant it is to $T$. Concept vector generation algorithm is depicted in the following:

**Algorithm: concept vector generation**

**Input:** page text $T$ extracted from web pages

**Output:** concept characteristic vectors $(O_1, O_2, ..., O_k)$

**Begin**

1. LoadMyDictionary(); //load words segmentation dictionary in a certain field
2. analyzer.segment($T$); //segment to get $T_i(T_i \in T, 1 < i < n)$
3. For each $T_i$ of $T$ do
   4. $O_i$ = MatchOntology($T_i$); //match between $T_i$ and concepts in ontology base
      5. If $\text{Instance.hasKey}$ not including $O_i$ then
         6. For each $O_j$ in hasKey do //check repeated concepts
            7. If $O_i$ and $O_j$ are synonyms then
               8. Exit
            9. Else
               10. $\text{Instance.hasKey}.O_i$.counter ++;
            11. End if
         12. Else
            13. $\text{Instance.hasKey}.O_i$.counter ++; //exist repeated concept
         14. End if
      15. Else
         16. $\text{Instance.hasKey}.O_i$.counter ++; //accumulate the counter
      17. End if
   18. End for

19. $\text{Instance.hasKey.weightVector}(O_1, ..., O_k)$; //compute weightings of the concept vectors

**End**

After the above three steps, concept characteristic vectors are constructed, as well as non-structured web pages are converted into structured semantic information that computer can understand.

### 4. Practical Experiments

The practical experiment was done on NetEase. We selected a content web page sample from the website, whose URL was http://auto.163.com/08/0219/08/45264FBS00081S9K.html, as shown in Figure 6.

![A web page sample](image)

**Figure 6.** A web page sample
At first we used our web page pretreatment module to remove and filter noises in the web page and created its DOM tree, then we obtained its page text using the page text extraction program we designed, as shown in Figure 7.

![Figure 7. Page text extracted from the web page](image)

In order to extract the concept characteristic vectors of the web page text, we constructed an ontology model of automobile information domain by using Uschold&King’s skeleton method, and using the programs developed for semantic information processing we obtained the final results, as shown in Table 1. Thus, we got the major idea of the web page combining statistics method and semantic analysis method based on ontology.

<table>
<thead>
<tr>
<th>No.</th>
<th>Concept characteristic vectors</th>
<th>Weightings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>丰田</td>
<td>0.247531</td>
</tr>
<tr>
<td>2</td>
<td>新皇冠</td>
<td>0.163189</td>
</tr>
<tr>
<td>3</td>
<td>皇冠</td>
<td>0.302693</td>
</tr>
<tr>
<td>4</td>
<td>车型</td>
<td>0.090682</td>
</tr>
<tr>
<td>5</td>
<td>混合动力车</td>
<td>0.074259</td>
</tr>
<tr>
<td>6</td>
<td>Royal</td>
<td>0.026975</td>
</tr>
<tr>
<td>7</td>
<td>Athlete</td>
<td>0.026975</td>
</tr>
<tr>
<td>8</td>
<td>刹车</td>
<td>0.060237</td>
</tr>
<tr>
<td>9</td>
<td>安全气囊</td>
<td>0.038812</td>
</tr>
</tbody>
</table>

5. Conclusions

This paper presents a new method about the web information extraction based on ontology. Compared with other web information extraction technologies, the introduction of ontology will improve information extraction accuracy and the recall rate significantly. However, a general-purpose ontology can’t be constructed for the moment, we can only construct specific field ontology semi-automatically, and more problems about web information extraction based on ontology still need further discussions.

6. Acknowledgement

This work was supported by the National Natural Science Foundation of China (No.70971124), Shandong Soft Science Research Program (No.2010RKBG1147), Shandong Province Higher
Educational Humanities and Social Sciences Research Project (No.J10WG64) and Shandong Province Housing and Urban-Rural Construction Science Research Program (No.2011RK041).

7. References