Data Extraction Based on Page Structure Analysis

Yichao Ren¹, and Jiayin Tian².

¹College of Information Engineering, Communication University of China, Beijing, China

² College of Information Engineering, Communication University of China, Beijing, China

Abstract. The information we need has some confusing problems such as dispersion and different organizational structure. In addition, because of the existence of unstructured data like natural language and images, extracting local content pages is extremely difficult. In the light of the problems above, this article will apply a method combined with page structure analysis algorithm and page data extraction algorithm to accomplish the gathering of network data. In this way, the problem that traditional complex extraction model behave poorly when dealing with large-scale data is perfectly solved and the page data extraction efficiency is also boosted to a new level. In the meantime, the article will also make a comparison about pages and content of different types between the methods of DOM structure based on the page and HTML regularities of distribution. After all of those, we may find a more efficient extract method.

Keywords. Page Structure Analysis, Data Extraction, JSON

1 Introduction

The content on Internet is all half-structured data, which contains structural data and non-structural data like natural language and images. Web has already developed to a giant source of information. What's more, Web information extraction with optimistic application prospect has certainly become a research hotspot these years. Exactly because of the manifold information on the network and half-structured web data, it is difficult for us human to search useful information among the sea of data. Not even to think about popular application programs, for they are unable to analysis diverse information on the Internet directly. So as to enhance the availability of network data, web information extraction shows up. The technology makes it easy to extract the information on the web page in a more structured way. These useful information could be helpful in data analysis system, batch query system and so on.

The information we need has some confusing problems such as dispersion and different organizational structure. In addition, because of the existence of unstructured data like natural language and images, extracting local content pages is extremely difficult. In the light of the problems above, this article will apply a method combined with page structure analysis algorithm and page data extraction algorithm to accomplish the gathering of network data[1]. In this way, the problem that traditional complex extraction model behave poorly when dealing with large-scale data is perfectly solved and the page data extraction efficiency is also boosted to a new level. In the meantime, the article will also make a comparison about pages and content of different types between the methods of DOM structure based on the page and HTML regularities of distribution[2]. After all of those, we may find a more efficient extract method[3].

2 Data Extraction Based on Page DOM Structure Analysis

The technology uses hierarchical relation of label to combine the content in webpage. As a result, an orderly structure is built up. Then, the technology gets useful data by matching rule of regular express[4].

If the language matching the structure of data could not be described by context-free grammar of global consistent query accurately, and there is no locally uniformly context-free grammar of graduation, such as natural language and image, could be obtained through any ordered partition about data, then the data is non-structured. If the language matching data structure could not be described by context-free grammar of global consistent query accurately, but there is a well-organized division of data in which the local consistency of context-free grammar of Partition No. $i+1$ could be concluded by the clue of the semantic information of the foregoing partitions, then the data is semi-structured[5]. Directly, data extraction is a process during which we find the useful content from non-structured or semi-structured data and then extract the data to make it structured and clear.

The algorithm based on the structure of tree depends on the structure of label in text. However, standard has little requirements on grammar, and different webpages that come from different designers do not share a common executable power of standard. For example, labels often do not match the content. Now, almost all of Web programming regards HTML of webpage as a document tree. From the observation of DOM document tree, we know that many labels can be deleted in advance and there is not a bit influence on main data content of HTML. DOM document tree is the structure tree of HTML which defines many normal connectors that have specific access to webpage labels. DOM is a connector that is completely independent of different languages and platforms. Different languages and platforms could also have the chance to access and operate the structure, data and even layout of DOM text. Here comes several frequently-used points of HTML DOM:

Get Element By Id(id): obtain node(element) with specified ID

Append Child(node):insert new child node(element)

Remove Child(node):delete child node(element)

Several frequently-used properties of HTML DOM:

Inner HTML: textual value of node(element)

Parent Node: parent node of node(element)

Child Nodes: child node of node(element)

Attributes: attribute node of node(element)

After normalizing, HTML source code is parsed to DOM tree structure. Each label of HTML corresponds to each node of DOM tree, and the property of node is exactly the property of label. HTML node of DOM tree has two child node: Head and Body. Head label defines the head of document and it is the container of all head elements[6].

Element in the head could be adopted in script, indicate browser to find style sheet, provide meta-information and so on. The head of document describes property and information of the file, including title, location in the web and connection with other documents. The data contained in the head would not be revealed as normal content in most documents.

Element of body defines subject of the whole document. Body element contains all content of the file, such as text, hyperlink, image, form, list and so on. The analysis of DOM tree is mainly in body elements and all child node of it. Structurally, the child tree of DOM tree that body lies contains all kinds of structure information, for example, XPath property of label A, class property of DIV label, label FROM and the like.

HTML label is of nested feature, so HTML webpage has a hierarchical structure. Many methods parse HTML document through DOM tree, operating every single node(HTML label) to extract text or other feature of the node. If a leaf node does not contain text, then we delete the node expediently. Analyzing the link between regions in the webpage and semantic connotation between hierarchical structure by page structure, then independent meaning unit is determined.

Applying algorithm to extract webpage content on DOM parse tree. Below is specific description of the algorithm:

The algorithm expresses the node which contains 'w'-the number of information, remarked 'n', remarked 'p' on the parent node by the form of tetrad-(s,w,n,p). Gaining the end of extracting information of content by finding a set of nodes sequence in tree T.

The tetrad-(s,w,n,p) works like this. Location number-s shows the location sequence of nodes,

counting from left to right, up to down. As a result, the 's' value of root node is 0. The quantity of information-'w' shows the number of characters in both Chinese and English that included in the nodes of the tree. The flag value of the exact node is represented by 'n', while the flag value of parent node of the node in the above is represented by 'p'.

We are searching for a set of nodes sequence in which node tag-'n' and parent tag-'p' of every node array must equal to each other and the distance of any two nodes-'Xi' and 'Xj' in the tree equals to different value of node-'j-i'. What's more, 'i' and 'j' mean the location number of node.

Then, we handle the sequence by getting set like the following steps:

Do away with node sequence remarked 'a' or node sequence of which node information-'w' is less than 'q'. What calls for special attention is that the value of 'q' is 16 in our system.

Combine or filter the sequences with only one node in the sequence set. Inspect parent node of the node in all single node sequence with algorithm, then combine the nodes that the parent node of which comply with the requirements for algorithm to generate a new queue. If the node that conforms to the condition is not node Max, then do the step of filtering and deleting. If the node is node Max, then reserve it. When the filtration is over, export all node information of the remaining nodes sequence. What we get is exactly the content information of extracted webpage.

The label of HTML could be divided two types based on the difference of function of webpage label. One type is structure label aiming to build frame of webpage, while the other is layout label specifically for modifying text of webpage. We only care about the structural distribution of webpage, so we need to filter the layout label that modifies webpage text, such as font label, text layout label, , <small>, <big>, , <i>, , <u>, text style label<style> and so on. Then, we only consider structure label when we remark node with quadruple notation, such as <p>, <div>, <table>, <tr>, <td>, , and so on.

Subject data extraction based on webpage mainly filters the webpage in the process of judging webpage

and subject. However, the difference of webpage and traditional text information is the webpage is semi-structured. Webpage expresses different meaning of text through different labels, and what we need is the useful information in the page while analyzing webpage and subject semantics, such as text, title, link and so on. So, we have to extract these information from webpage and represent text with the meaning of label when analyzing the relevant semantics[7].

3 Data Extraction Based on HTML Regularities of Distribution

The main idea of the method is to extract body content of the webpage step by step. At first, webpages of common similar structure need to be blocked. In other word, the whole webpage will be divided to two parts-head and body. Then, useless label elements and the related content will be deleted by analyzing HTML label semantic in the two domain blocks[8].

In webpage structure generated by HTML, XHTML, XML auto format, there are <head></head> and <body></body> which mean head block and body block.

Take out the content of body block. Body block format is as below:

<body>

The content of body block:

- (1) The content of body block:
- (2) Webpage text content
- (3) Others

</body>

Delete useless information in the body block. Body blocks of movie reviews page in Douban consists of text in which contains series of property with controlling power like size, color, font and so on. Then, delete content irrelevant to text content in body block.

Delete information irrelevant to text content in body block. It always occurs that there are some word links and image links which are irrelevant to the text content. In order to extract text without irrelevant links, all irrelevant links need to be cleared.

We could get webpage text content smoothly through the method of deleting HTML element based on regional block. It satisfies the need of the article, and hit the mark of extracting text content.

4 Text Extraction Experiment

Similar with commonly used data extraction, data extraction of user leading webpages, such as post bar, forum, Weibo, has to follow the design principle of common data extraction like flexibility and robustness. Because of special need of system, data extraction part is different from common data extraction.

We take user reviews information in Douban for example to carry on the experiment. When users are surfing on the movie section, they can choose an interested movie section. Then, simply click the area to enter the page of the selected section. What shows up is relevant content about the section. When users choose a specific movie label, the page changes to a page of the exact movie. So users could comment in the page. Posts page is regarded as leaf node of forum.

The experiment extracts text in HTML source code by regular expression. Analyzing the code with beautifulsoup in Python.

Another easier way is to match text content in original HTML source code by regular expression directly. The rule of replacement is '`<(\s|.)*>`'. '`\s`' means blank character which contains blank character, CRLF and tag symbol. '`|`' means the front one or the latter one is chosen. '`.`' means any symbol, so '`\s|.`' means that match all arbitrary character. '`*`' matches none or multiple symbol that has showed up before. Thus, '`(\s|.)*>`' means that any symbol shows up any times which equals to any alphabetic string. So, we can conclude that regular expression '`<(\s|.)*>`' means any alphabetic string that starts with '`<`' and ends with '`>`'. The alphabetic string is exactly HTML label. We replace the text according to matching rule to filter the HTML label in HTML source code.

Fig.1. Data Etraction

5 The Text Comment Is Put Out As Data of JSON Format

JSON as a standard module could accept Python data structure and turn them to a representation of string. This process is called serialize. The types that JSON code supports contains none, bool, int, float and str. There is no doubt that the types also contains lists, tuples and dictionary that of basic type. As for dictionary, key is assumed as string. It is necessary to note that any non-string key would be transferred to string during coding process. To confirm to JSON standard, only Python list and dictionary should be coded.

In different languages, a collection of name/value pairs is unscrambled as object, record, structure, dictionary, hash table, keyed list or associative array. In most languages, an ordered list of values is unscrambled as array.

These are usual data structure. In fact, most modern computer languages support them in some form which makes the exchange of a data structure in programming language based on these structure in the same way possible.

The simple data types that put out through encode and repr() which represents the original way share a common result. However, several types of data change during the process. For example, the tuples mentioned in the above are converted to list. During the programming process of JSON, there is a conversion between Python which is the original type and JSON. Some basic types of Python such as tuple turn to list. When the object is turned back to python type, it doesn't come with the change that list type turns to tuple one. In the meantime, the format of the code changes to unicode.

The method of json.dumps() reverse a object back which is of str type and called encode json.

```
encodedjson = json.dumps(obj)
```

An object of python could not convert to json type directly. The only way to finish the conversion is to change the object to dictionary first, and then change it to json. As for json, the conversion acts as the same way that the target is changed to dictionary as a transfer station[9].

As for the text comments that we have already got through the methods above, we transfer them to json.

First of all, we create a method, analyze data, rewrite default method of JSONEncoder, and convert object to dict.

```
class userEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, user):
            return {
                'name': obj.name,
                'comment': obj.comment
            }
        return json.JSONEncoder.default(self, obj)
    else:
        return obj
```

The Encode process is to convert python object to json object. There are two frequently-used functions, one is dumps and the other is dump. The only difference between them is that a document flow of fp is generated during the conversion of dump while during the conversion of dumps, a string is generated instead.

When JSON deals with Chinese string, the string is showed as unicode. As a result, unicode usually acts as intermediate code during a conversion. The process works as this: decode string in other code to unicode, and then, transfer unicode to another code. The role of decode is to convert strings in other code to unicode. For example,

`str1.decode('gb2312')` means convert `str1` coded by `gb2312` to unicode. The rule of encode is convert string in unicode to strings in other codes. Here comes another example, `str2.encode('gb2312')` means convert `str2` to string coded by `gb2312`. Therefore, we have to make it clear before we start conversion.

We define the variable, convert dictionary to object in JSON, put out data of JSON. The result is as below.

[illegible]

Fig.2.JSON Transmission Format

The reason responsible for the result is that `json.dumps` generates matching character encoding instead of original character towards non-ascii character in default occasion.

If the original character is required, just handle as below:

```
print json.dumps(js, ensure_ascii=False)
```

6 Conclusion

HTML language is semi-structured. It lacks of semantic to describe information. So, when it comes to analyzing webpage structure by DOM tree, most of the methods are studying how to use the semantic information to design data waiting to be extracted by regular expression with directed attention. Then, the detail data could be extracted from the webpage.

Web information extraction method based on HTML regularities of distribution or the relationship between the tags is appropriate for simple webpages with regular structure. The method does good at this kind of webpage. And the method could recognize text content in unconventional web structure correctly. Then, the exact text content could be extracted.

JSON is a simple format of text same as XML. Relative to XML, JSON is more readable and easier to check by users. At the grammatical level, the difference between JSON and other formats is characters that separate data. Characters that separate data in JSON are confined to single quotes, parenthesis, bracket, brace, colon and comma. At first glance, the advantage of using data delimiter in JSON is not obvious enough. However, there is an essential reason that makes JSON special. Data delimiters in JSON simplify data access which will open an easier access path than DOM. Another advantage of JSON is its feature of not long. In XML, it is necessary to open and close the tag which leads to the satisfaction of compliance of tag. But in JSON, all these requirements would be satisfied only through a simple comma. In the process of data exchange that contains hundreds of fields, traditional XML tag would lengthen the time of data exchanging.

Acknowledgment

The work on this paper was supported by the open project foundation of Key Laboratory of Performing Art Equipment System Technology of Ministry of Culture, P. R. China.

References

- [1] Hong, Haikun, et al. "Web Data Extraction Based on Tree Structure Analysis and Template Generation." International Conference on E-Product E-Service and E-Entertainment IEEE, 2010:1-5.E.
- [2] Ma, Anxiang, et al. "Web Data Extraction Based on Structure Feature". Applied Informatics and Communication. Springer Berlin Heidelberg, 2011.
- [3] Hongyao Chang, "Research on The Body of Web Pages Based on Labelling Analysis". Chongqing University, 2015.
- [4] Wen Zhao. " Research on Web Page Structure Mining for Information Extraction". Artificial Intelligence and Recognition Technology, 2008, 1(1):126-129.
- [5] Jiang Liu. "A survey on sparse subspace clustering." Harbin Institute of Technology, 2012.
- [6] Wang, Jiying, and F. H. Lochovsky. "Data extraction and label assignment for web databases." International Conference on World Wide Web ACM, 2003:187-196.
- [7] Cormier, Michael. "Computer vision-based analysis of web page structure for assistive interfaces." Web for All Conference ACM, 2016:24.
- [8] Chau, Michael, and H. Chen. "A machine learning approach to web page filtering using content and structure analysis." Decision Support Systems 44.2(2):482-494.
- [9] Cesarini, Francesca, et al. "Page Classification for Meta-data Extraction from Digital Collections." International Conference on Database and Expert Systems Applications Springer-Verlag, 2015:82-91.