ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Parsing of HTML Document

Pranit C. Patil[1], Pramila M. Chawan[2], Prithviraj M. Chauhan[3]

Abstract: **The Websites are an important source of data now days. There has been different types of information available on it. This information can be extremely beneficial for users. Extracting information from internet is challenging issue. However the amount of human interaction that is currently required for this is inconvenient. So, the objective of this paper is try to solve this problem by making the task as atomic as possible.**

Existing methods addressing the problem can be classified into three categories. Methods in the first category provide some languages to facilitate the construction of data extraction systems. Methods in the second category use machine learning techniques to learn wrappers (which are data extraction programs) from human labeled examples. Manual labeling is time-consuming and is hard to scale to a large number of sites on the Web. Methods in the third category are based on the idea of automatic pattern discovery. For extracting information from web firstly we have to determine the meaningfulness of data. Then automatically segment data records in a page, extract data fields from these records, and store the extracted data in a database. In this paper, we are given method for an extracting data from SEC site by using automatic pattern discovery.

Keywords- Parsing engine, Information Extraction, Web data extraction, HTML documents, Distillation of data.

1. INTRODUCTION

As the amount of information available from WWW grows rapidly, so extracting information from web is a important issue. Web information extraction is an important task for information integration, because multiple web pages may

Pranit C. Patil (M.Tech. Computer, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai-19, Maharashtra, India)

P.M.Chawan (Associate Professor, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai-19, Maharashtra, India)

Prithviraj M.Chauhan(Project Manager, Morning Star India Pvt. Ltd., Navi Mumbai, Maharashtra, India)

may present the same or similar information using completely different formats or syntaxes, which makes integration of information a challenging task. How to improve the performance in the
information retrieval its has become an urgent problem needed to be solved, so that in this paper we are given a method for extracting data. The rapid increase in the number of users, the volume of documents produced on the Internet is increasing exponentially. Users wish to obtain exact information from this deluge of available information. Furthermore, users wish to gain other useful information by processing these data. Text mining is one method that can meet these requirements.

In this paper, we are extracting data from SEC (Securities Exchange Commission) site. So there are different forms are submitted by various company on SEC site. These forms are in various formats ie. HTML, text. We are extracting data from SEC site for Financial Advisor Company. To entering the data into the financial companies database, a employee of the financial company manually need to read the forms and enter it into the interface of company. This activity is more and more complex. To overcome the above mentioned problems, an automated system for inserting the required data in the database of the companies is proposed in this paper. This system will parse the HTML pages from the SEC website.

The proposed system will use the standard parser Libraries available for JAVA. The challenging job in this system is to develop such an intelligence that will identify various patterns frequently occurring in the forms and also the patterns that are possible. The information contained within the form is either in the plain text format or in the tabular format. The system will also be able to identify the exact row and column for identifying the required information.

This paper is organized as follows. We begin with section 2 we gave related work of this topic. In Section 3 we given analysis of SEC form. In Section 4 we given design work for extracting data.

2. RELATED WORK

The World Wide Web is an important data source now days. All useful information is available on the websites. So extraction of useful data from that data is an important issue. There are three approaches are given for extracting data from web.

The first approach by Hammer, Garcia Molina, Cho, and Crespo (1997) [1]. is to manually write an extraction

program for each web site based on observed format patterns of the site. This manual approach is very labor intensive and time consuming and thus does not scale to a large number of sites.

The second approach Kushmerick (2000)[2] is wrapper induction or wrapper learning, which is currently the main technique . Wrapper learning works as follows: The user first manually labels a set of trained pages. A learning system then generates rules from the training pages. The resulting rules are then applied to extract target items from web pages. These methods either require prior syntactic knowledge or substantial manual efforts. An example of wrapper induction systems is WEIN by Baeza Yates (1989).

The third approach Chang and Lui (2001)[3] is the automatic approach. Since structured data objects on the web are normally database records retrieved from underlying web databases and displayed in web pages with some fixed templates, automatic methods aim to find patterns/grammars from the web pages and then use them to extract data. Examples of automatic systems –IEPAD by Chang and Lui (2001), ROADRUNNER by Crescenzi, Mecca, and Merialdo (2001).

Here we gave a automated extraction method for data extraction from the web, we also give an algorithm for extraction purpose. When manually extracting data from the SEC web site, then it is a more time consuming and this activity is also the more and more complex. So, for overcome this problem we are given a system that extracting data from SEC website.

There are some challenges in extracting data from SEC wbsite these challenges are given as,

- How to parse HTML documents? : There was no good reason and sufficient knowledge or existing application we could figure out for parsing html documents. Data points were scattered all around the document and was available in different region namely table section and non-table section of the document. Parse html documents into two different section, table section and non-table section and apply some pattern matching or other rules and algorithm for parsing these sections for retrieving data points.
- How to parse Text documents?: The system have to parse the text documents because some of the form which submitted in SEC are .txt format. So for extracting data from text format documents it is important to identify the String patterns.
- Processing Engine and Performance Issue: Any available parsing techniques will do. Free to use own implementation and processing style. In the engine using regular expression or natural language processing.
- Algorithmic Engine: There are several ways of processing documents for Information Extraction namely Natural Language Processing tools, compiler phases like generating tokens, syntax and semantic analysis, lexical analysis, or some other parsers.

Non-table section of document was processed using matching set of keywords and some logic of extracting particular row and columns having data points.

3. ANALYSIS OF SEC FORMS

We have to extract the information from the SEC (Security Exchange Commission). So, its important to study the SEC forms. The document available in two different region namely table section and non-table section. In this paper we are analyzing the only table section. We extracting data from the tables of the SEC documents. We began our Project Analysis by going through several SEC files, which are available on WWW.sec.gov.in. There are number of forms filing is present on this SEC web site, we are analyzing DEF 14A, 10K, 8K for this project. The various fields that are useful for the financial advising companies are available in the above forms.

The analysis of tables is based on the following assumptions. a) The table is organized row-wise and the rows of the tables are classified into heading row and content rows. The first row of the table is always the heading row and rest of the rows contains the data. b) Tables that we are considering here are one-dimensional tables. Additionally, the one-dimensional tables can be further classified as row-wise, column-wise or mix-cell tables.

3.1 Identify Head components:

The table is first analyzed to distinguish between the header row and the content rows. The header row defines to be a row that contains global information of all content rows. Primarily a heading explains the quantities in the columns. It is not that much easy to distinguish between the header rows and the content rows. In general, the header row can be identified by following rules:

1. The content of the header cells is not a quantity. In other words, from the programming view, the header cells should not contain the numbers and it should contain strings.
2. The header rows are normally the top most rows of the table. Other top most rows may contain spaces or some may contain Non-breaking Spaces (in HTML).
3. The header row is visually different from content rows. This happens when the table contents are graphically coloured or decorated by fonts and other styles.
4. The header row contains significantly fewer cells per row

However the above said rules are not complete set of rules to distinguish the header row from content rows. Some more rules are to be designed for complex tables. Also, sometimes in few cases it depends on the format of tables.

Most of the time some tables are in standard format and the possible cell contents of the header row are known in advance. This is the case for the tables given on the web pages

of filings on SEC website [10]. In such cases, it becomes very easy to identify the header rows, which can be identified by matching certain strings with the cell contents. The string to be matched should be the standard strings which are expected to occur in the header row. For Example, again consider the example of [10]. The DEF14A type of filings on SEC website contains a table named as "Executive Details" table. The minimum fields that are expected in this table are Name, Age, Position. In addition to these three fields they can give "Director Since" field. In such a case we can match the cell-contents of each row, with the expected strings. The point to be noted here is that, we are assuming a one-dimensional table which contents only a single header row. Thus there will be only one valid header row. Considering the second point, the header row of a table is mostly at the top most position of the table. Some rows that may be above the header rows may contain blank spaces or Non-breaking Spaces (in HTML). Such unnecessary rows should be neglected, which can be identified by scanning the cell-contents for spaces like Non-breaking Space. Our third rule compares the visual characteristics of the header row with that of a typical row. As stated in [11], the most reliable approach to distinguish the heading from content rows is the cell count. We can use the rule followed in [11] that if the cells count in the row being considered is equal or less than 50% of the average cell count, then it is a heading. Otherwise it is a content row. After identifying the heading row, we will keep the record of those cell-contents in a set or a collection. We will consider this information later to identify the attribute-value pairs from remaining rows.

3.2Row Span and Column Span :

For the purpose of data extraction from the table row span and column span is important. Row and Column spanning is often used in HTML in order to allow a cell to occupy more than one row or column. By default, a cell occupy (both row-wise and column-wise) only a single row or column. A positive integer is associated with the cell attributes which decides the number of rows or columns occupied by the cell. For example "rowspan=4" means the cell spans 4 rows consecutively, starting from the current column. For identifying the attribute-value pairs in a table where for some cells the value of the rowspan or colspan attributes is greater than "1", the simplest way is to duplicate the cell contents to each row or column it spans. For example, again we will consider the tables given in filings on SEC [4] website. If we consider the DEF14A filings on SEC, we can get a "Summary Compensation" table. For this table, the column is always labeled with the attribute "name". For a single name, usually there are more than one entries associated with that name attribute. This is because the value of the "rowspan" attribute of the cell containing "name", is more than one.

In analyzing the one-dimensional web tables, it is normally the case that the attribute and the corresponding values are placed in a single column. Thus, we can say that the collection of values from a single column represents the complete data set for the attribute to which the column belongs.

4. DATA EXTRACTION FROM SEC

In order to extract the data structure from a web-site, we need to identify the relevant fields of information or targets; to achieve that, there are five problems that need to be solved: localizing of HTML pages from various web databases sources; extracting the relevant pieces of data from these pages; distilling the data and improving its structure; ensuring data homogeneity (data mapping); and merging data from various HTML pages into a consolidated schema (data integration problem).

4.1Find the files from SEC site:

There are several types of forms are submitted in SEC we are using only HTML and text files. We have to identify the document type of the form. Then we have to identify that the form is DEF 14, 10K or 8K, which we are going to parse in the system.

4.2 Data extraction from HTML pages:

Once the relevant HTML pages are returned by various web databases, the system needs to localize the specific region having the block of data that must be extracted without concern for the extraneous information. This can be accomplished by parsing each HTML page returned from different databases. Java has a parser method that parses HTML pages, so this method was used for parsing HTML pages. The foreign database which given in above diagram is SEC database.

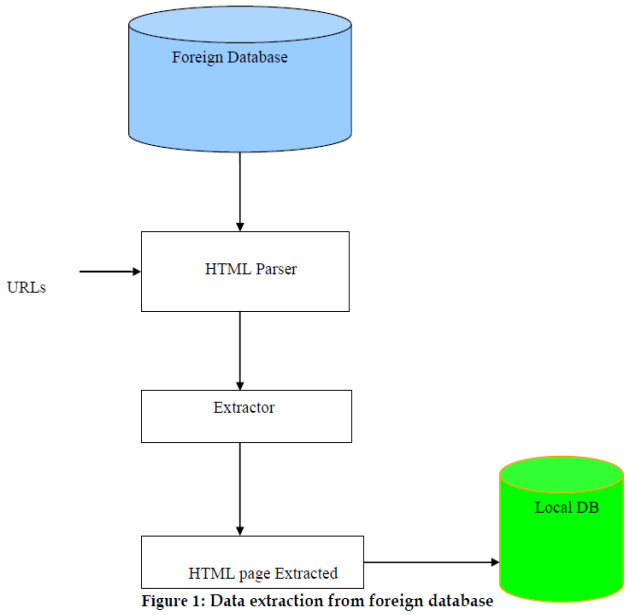

Figure 1: Data extraction from foreign database

The above diagram shows the extraction of data from HTML pages. We are going to use Jericho Html parser in the project for extracting data from the HTML pages.

4.3 Distillation and improving data:

The data extracted from the response pages returned by various database sources might be duplicated and unstructured. To handle these problems of duplication and lack of data structured, the data extracted needs to be filtered in order to remove duplication and unstructured elements, and then the data result must be presented in structured format. This is accomplished by passing each HTML data extracted through a filtering system; this processes the data and discards data that is duplicated.

4.4 Data Mapping:

The data fields from various SEC database sources may be named differently. It will also having the different types of forms. A mapping function is required for a standard format and improves the quality of HTML data extraction. This mapping function takes data fields as parameters and feeds them into a unified interface.

4.5 Data Integration:

The final step in the problem of data extraction is data integration. Data integration is the process of presenting data from various Web database sources on a unified interface. Data integration is very crucial to the users for the reasons that users want to access as much information as possible in less time. The process of querying one airline at a time is time consuming and sometimes very hard; especially when the users do not have enough information about airline sources.

5. HTML Data Extraction

The SEC files are in HTML pages. When we have to extract data from HTML pages then there will be different steps for extracting data from HTML pages. The different steps for extracting HTML data given as follows:

5.1 Preprocessing HTML pages:

Once a query is sent by a user or Financial Advisor company to a SEC databases, the HTML pages are returned and passed to the parser system in order to analyze and repair the bad syntax structure of HTML documents and then extract the required data on the page. The process is performed in three steps. The first step consists of the retrieval of an HTML document once this page is returned by the SEC database and possibly its syntax reparation. In the second step, is responsible for generating a syntactic token parse tree of the repaired source document; and the third and last step is responsible for sending the HTML document to an extractor system for extracting the information that matches the user need.

5.2 Repair Bad Syntax:

The SEC files are in HTML format having the bad syntax so it is important to remove bad syntax. It repairs bad HTML syntax of the document by inserting missing tags, and removing useless tags, such as a tag that either starts with < Pr which is an end tag that has no corresponding start-tag. It also repairs end tags in the wrong order or illegal nesting of elements. It describes each type of HTML error in a normalization rule. The same set of normalization rules can be applied to all HTML documents.

5.3 Parsing:

The parsing of HTML document to the XML output is as shown in figure below. For the parsing purpose of data we are parse the document from JAVA parser. Then we differentiate both the table and non-table section. Then we pass down the table section using table pattern matching engine.

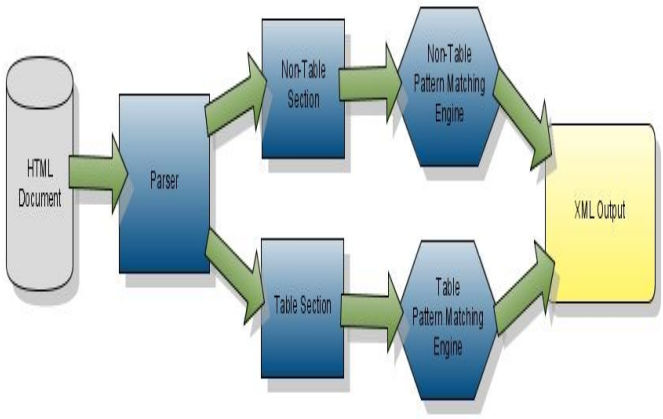

Figure 2. Workflow diagram

5.4 Interpretation Algorithm:

A simple table interpretation algorithm is shown below. We assume that there are x rows and y columns. Also, let celli,j denote a cell in ith row and jth column.

1. Considering the basic condition, if there is only one row, then we can say that the table does not contain any data. If contains the data, because only one row is present it does not contain the header-row to identify the attributes. Such table needs to be discarded, because we can't extract the attribute-value pairs from it.
2. If there are two rows then the problem becomes very easy. The first row is treated as header row and the second one as the content row. Otherwise,
we start the cell similarity checking from the first row in step 2.
3. For each row i ($1 \leq i \leq x$), compute the similarity of the two rows i and i+1. If i = x, then compute the similarity of i and i-1 and then stop the process.
4. If the ith row is empty, then go for the next pair of rows, i.e. i = i+1.
5. If the ith and (i+1)th row are not similar and $i \leq (x/2)$, then the ith row is treated as header row, in other words the ith row contains the attribute cells. Store the labels of attributes in a

list and index each label with position in row. Count the number of valid data cells in header row. After identifying the ith row as header row, we will continue to find the content rows only.
6. If the ith and (i+1)th row are similar and also both the rows are non-empty, then count the number of valid data cells in both rows. If both the counts are equal or approximately equal to the valid data cells count of header row, then both rows are treated as the content rows. Store the cells content of each row in a list indexed with their position in row.

6. CONCLUSION

In this paper we given an efficient way to extracting information from the SEC site. Firstly, we analyze all the formats of the forms of SEC site. Then we studied head components and also rows and column span.We given the five steps for the extracting information from SEC site. First find the type of SEC document, Data extraction of web, Distillation and data improving, Data mapping, Data Integration. Then we given HTML data extraction workflow diagram. In this first preprocessing HTML pages, repair bad syntax then parsing the HTML document. Then we also given the table interpretation algorithm. We used the cues from HTML tags and information in table cells to interpret and recognize the tables.

REFERENCES

[1] D.W. Embley, Y. Jiang, and Y.K. Ng, "Record-Boundary Discovery in Web Documents," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 467-478, 1999.

[2] B. Liu, R. Grossman, and Y. Zhai, "Mining Data Records in WebPages," Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 601-606, 2003.

[3] H.H. Chen, S.C. Tsai, and J.H. Tsai, "Mining Tables from Large Scale HTML Texts," Proc. 18th Int'l Conf. Computational Linguistics, July 2000.

[4] D. Buttler, L. Liu, and C. Pu, "A Fully Automated Object Extraction System for the World Wide Web," Proc. 21st Int'l Conf. Distributed Computing Systems, pp. 361-370, 2001.

[5] M. Hurst, "Layout and Language: Beyond Simple Text for Information Interaction—Modeling the Table," Proc. Second Int'l Conf. Multimodal Interfaces, 1999.

[6] G. Ning, W. Guowen, W. Xiaoyuan, and S. Baile, "Extracting WebTable Information in Cooperative Learning Activities Based on Abstract Semantic Model," Proc. Sixth Int'l Conf. Computer Supported Cooperative Work in Design, pp. 492-497, 2001.

[7] V. Crescenzi, G. Mecca, and P. Merialdo. Automatic web information extraction in the roadrunner system. In Proceedings of the *International Workshop on Data Semantics in Web Information Systems (DASWIS-2001), 2001*.

[8] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large web sites. In Proceedings of the *27th Conference on Very Large Databases (VLDB)*, Rome, Italy, 2001.

[9] C. H. Chang and S. C. Lui. IEPAD: Information Extraction based on Pattern Discovery. In *10th International World Wide Web Conference (WWW10)*, Hong Kong, 2001.

[10] www.sec.gov

[11] Yingchen Yang, Wo-Shun Luk, School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada, *"A Framework for Web Table Mining", WIDM'02, November 8, 2002*, McLean, Virginia, USA.

[12] http://jericho.htmlparser.net/docs/index.html

[13] http://htmlparser.sourceforge.net/