ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Automated Data Extraction for Unsupervised Web Documents

Roselin Paul

M.Tech Student, Dept. of C.S.E, VJCET, Vazhakulam, Kerala, India

ABSTRACT: Data extraction is the process of retrieving data out of data sources for further data processing or data migration. Data mining can be defined as a process of extracting patterns and data from the internet. Wrapper in data mining is a program that extracts content of a particular information source and translates it into a relational form. Web data extractors are used to extract data from web documents in order to feed automated processes. The general problem of data extraction is to extract relevant data from the encoded web document. Many web data extractors rely on extraction rules which can be classified into supervised and unsupervised technique. Supervised techniques requires the user to provide samples of the data to be extracted while unsupervised techniques is the techniques that learn rules that extract as much prospective data as they can and the user then gathers the relevant data from the results. Proposed system is an unsupervised technique which deduces the relevant data from XHTML encoded Websites. Proposed algorithm takes a set of template-generated pages as input and deduces the unknown template used to generate the pages and the values encoded in the pages are extracts as output.

KEYWORDS: Web data extraction, automatic wrapper generation, wrappers, unsupervised learning

I. Introduction

World Wide Web is a vast and rapidly growing source of information. There are many web sites that have large collections of pages containing structured facts, i.e., data having a structure or a schema. These pages are typically generated dynamically from an underlying structured source like a relational database. An example of such a collection is the set of book pages in Amazon. The data in each book page has the same schema, that is each page contains the title, list of authors, price of a book and so on. This method studies the problem of automatically extracting structured data encoded in a given collection of pages, without any human input like manually generated rules or training sets.

Extracting data from the web is a process in the field of data extraction. Internet pages in html, xml, etc are considered an unstructured data source due to the wide variety in the code, styles, and of course exceptions and violations of standard coding practices. Due to this variety, extracting data from the web is a highly customizable process depending on the specific source of information one is trying to retrieve. Web data extraction involves the process of extracting web pages of text-based markup language such as HTML and XHTML which usually contain a useful information. Thus data extraction is taking an unstructured form of data and parsing that information into a structured data set. Data mining can be defined as a process of extracting patterns and data from the internet. This process is quite important as data mining is increasingly becoming popular tool in creating models and decision making. Many web data extractors rely on extraction rules, which can be classified into ad-hoc or built-in rules. The costs involved in hand crafting ad-hoc rules motivated many researchers to work on proposals to learn them automatically using supervised techniques, i.e., techniques that require the user to provide samples of the data to be extracted, aka annotations or using unsupervised techniques, i.e., techniques that learn rules that extract as much prospective data as they can, and the user then gathers the relevant data from the results. Web data extractors that rely on built-in rules are based on a collection of heuristic rules that have proven to work well on many typical web documents.

II. Related work

Web information extraction is the task of identifying, extracting, and structuring relevant information from web documents in structured formats, e.g., tables or XML. The literature provides a collection of surveys on information extraction. Information extractors can be broadly classified according to whether they deal with free-text web documents or semi-structured web documents. The relevant information in a free-text web document is buried into sentences that are written in (telegraphic) natural language, whereas in semi-structured web documents it is buried into HTML scripts. Free-text web documents require natural language processing techniques to extract information from them. Contrarily, semi-structured web documents generally use HTML tags to typeset small pieces of information (attributes) in tables or lists. Independently from the kind of document on which it can work, the goal of an information extractor is to identify pieces of relevant information within a document and return them in a structured format. In Trinity[1] the technique works on two or more web documents generated by the same server-side template and learns a regular expression that models it and can later be used to extract data from similar documents. The technique builds on the hypothesis that the template introduces some shared patterns that do not provide any relevant data and can thus be ignored.

Roadrunner[2] extends traditional grammar inference to make it practically applicable to modern Web information extraction, thus providing fully automatic techniques for wrapping reallife websites. EXALG[3] works in two stages. In the first stage (ECGM), it discovers sets of tokens associated with the same type constructor in the (unknown) prototype used to create the input pages. In the second stage (Analysis), it uses the above sets to deduce the template. The deduced template is then used to extract the values encoded in the pages. In [4] the first module merges all input DOM trees at the same time into a structure called fixed/variant pattern tree, which can then be used to spot the template and the schema of the Website in the second module. Typical information extraction tasks focus on data regions and data records. That implies that as the complexity of typical web documents increases, information extractors have to analyze more and more irrelevant regions, which has an impact on both efficiency and effectiveness. This has motivated a number of authors to work on region extractors as a means to relieve information extractors from the burden.

III. PROPOSED ALGORITHM

Web data extraction involves the process of extracting web pages of text-based markup language such as HTML and XHTML which usually contain a wealth of useful information. Data mining can be defined as a process of extracting patterns and data from the internet. Figure 1 sketches the proposal, which takes a collection of web documents as input. The algorithm first finds the maximum length among the input web documents and sets a variable called s to max.

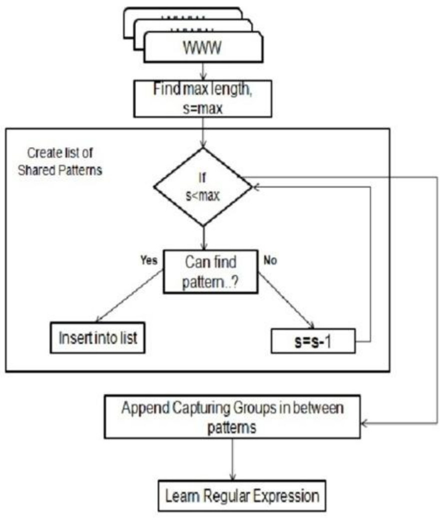


Fig. 1. General view of the proposal

```
Algorithm regularExpression(Html1,Html2):
        Find max Length from the documents
        if maxLength == 0:
                for count in range(0,len(dataList)):
                regExpr=dataList[count]+ alphabet[count]
                                break
                else:
                        Find SharedHtmlData(Html1,Html2)
                        Append into dataList
                Find UnsharedData(Html1,Html2)
                        Remove unshared data
                        Set the remaining as the input document
        Return
```

Fig. 2. The algorithm to learn regular expression

Starting with this node, the algorithm loops and searches for a shared pattern. If a shared pattern is found, then it is used to create new list with the shared pattern. Then find the unshared chunk from the beginning of a Text up to the first occurrence of opening tag in the input and then remove this unwanted stuff. Finally, get the new input documents after removing the unwanted chunk and the pattern. The input analyzed recursively in order to find new shared patterns and then induces new modified inputs. This process continues until the length of the input became zero. This is to

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

tokenize web documents into shared pattern and unshared chunk.

This built on the hypothesis that shared pattern are not likely to provide any relevant data and are part of template. Finally these shared patterns are organized into a list and later parsed it with capturing groups that represents the template that was used to generate the input documents. The capturing group represents the relevant data region in the web document and these capturing groups are represented as {A}, {B}, . . . , {F}.

A. Learn Regular Expression

A few training pages are used in machine learning to generate extraction rules. The system learns extraction rules from these pages. The rules are applied to extract items from other pages. After creating the shared pattern list an additional algorithm is used to learn the regular expression that represents the template used to generate the input web documents. This algorithm parses the shared data list and in every time it extracts a list item and append a fresh capturing group to extract the data that corresponds to the unshared chunk in between the patterns. The technique builds on the hypothesis that the template introduces some shared patterns that do not provide any relevant data and can thus be ignored. Fig.3 shows the flowchart of learning regular expression.

Web data extractors are used to extract data from web documents in order to feed automated processes. This algorithm works on two web documents generated by the same server-side template and learns a regular expression that models it and can later be used to extract data from similar documents. The algorithm first finds the maximum length among the input web documents and sets a variable called s to max. Whenever it finds a shared pattern, it partitions the input documents then removes the pattern from input document and save it in a data list and analyses the new input to find unshared chunk and then remove it. The result analyses recursively until no more shared patterns are found or the length of input documents become zero. Finally these shared patterns are organized into a list and later parsed it with capturing groups that represents the template that was used to generate the input documents. Automated extraction is an unsupervised technique, which learns a rule that extracts as much data as possible, give each capturing group a computer-generated label in between shared pattern and it is the responsibility of the user to assign a meaning to these labels. It is based on the hypothesis that web documents generated by the same server-side template share pattern that do not provide any relevant data, but help delimit them.

Fig.2 shows the algorithm to learn regular expression. Many web sites contain large sets of pages generated using a common template or layout. The steps below show the expansion of the algorithm. If the loop finishes and the current node was expanded, then the algorithm is executed recursively on the new level that have been added to the current node. Note that this algorithm does not use any tree structure, this is just the illustration of the algorithm.

- The algorithm works on two input documents. In the below sample web page lays out contain the author, title, comments. in the same way in all its book pages.
  1. <html><head><title>Results</title></head><body><h1>Results:</h1>C++<br/><b>Prata</b><br/>35.33<br/><br/>EffectiveC++<br/><b>Meyer</b><br/>33.95<br/></body></html>
  2. <html><head><title>Results</title></head><body><h1>Results:</h1>PHP<br/><b>Gilmore</b><br/>52.11<br/><br/>PHPsolutions<br/><b>Powers</b><br/>43.55<br/></body></html>
- When a shared pattern is found in the input, Algorithm creates the new list which stores the shared pattern.
- Then find the unshared chunk from the beginning of the remaining portion to the first occurrence of opening tag in the input and then remove this unwanted stuff.
- This loop iterates until the length of input document become zero.
- Whenever it finds the next shared pattern it will append to the existing shared data list. The final list of the above inputs is {<html><head><title>Results</title></head><body><h1>Results:</h1>; <br/><b>; </b><br/>; <br/><br/>; <br/><b>; </b><br/>; <br/></body></html>}
- The regular expression of above inputs are:
  {<html><head><title>Results</title></head><body><h1>Results:</h1> A <br/><b> B </b><br/> C <br/><br/> D <br/><b> E </b><br/> F <br/></body></html>}

B. Data Extraction

The World Wide Web is a vast and rapidly growing source of information. Most of this information is in the form of unstructured text and making the information hard to query. The problem of automatically extracting structured data encoded in a given collection of pages is solved without any human input like manually generated rules or training sets. Extracting structured data from the web pages is clearly very useful, since it enables us to pose complex queries over the data. Extracting structured data has also been recognized as an important sub-problem in information integration systems which integrate the data present in different web-sites. Therefore, there has been a lot of recent research in the database and AI communities on the problem of extracting data from web pages.

An important characteristic of pages belonging to the same site and encoding data of the same schema is that the data encoding is done in a consistent manner across all the pages. After learning the extraction rule from a set of web documents that were generated by the same server-side template, next is to extract data from similar web document. The algorithm proceeds as follows: It works on two inputs, that is the input web document and the regular expression. Regular expression represents the template of web document. Initially the data list is expected to be empty. The algorithm constructs its result by adding text to this parameters on each recursive invocation.

The technique builds on the hypothesis that the template introduces some shared patterns that do not provide any relevant data and can thus be removed. So the algorithm proceeded with the comparison of two inputs. As per the hypothesis, first search for the shared patterns in the documents and remove it from the template and then find the unshared chunk from the input document which is the relevant data. Whenever unshared stuff is find add it to the data list and this process continues until the length of the document become zero and returns the final data list. Fig.3. illustrates the flowchart of data extraction algorithm.

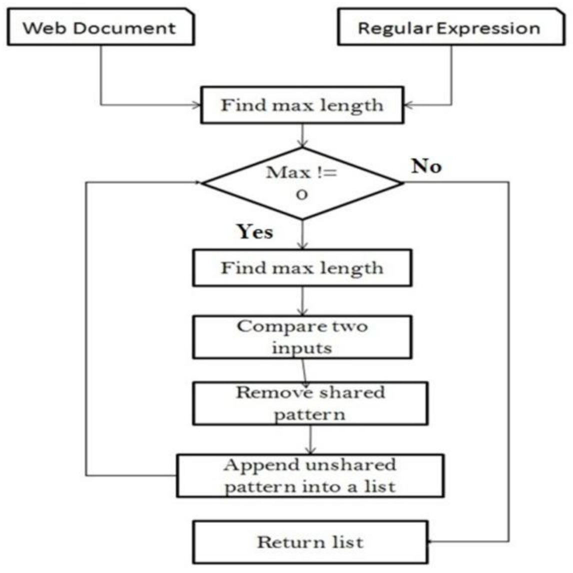
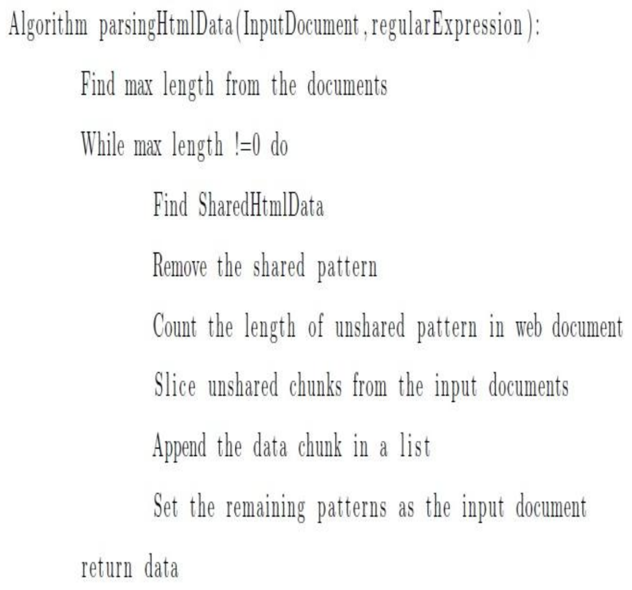


Fig. 3. Flowchart of Data Extraction



```
Algorithm parsingHtmlData(InputDocument,regularExpression):
    Find max length from the documents
    While max length !=0 do
        Find SharedHtmlData
        Remove the shared pattern
        Count the length of unshared pattern in web document
        Slice unshared chunks from the input documents
        Append the data chunk in a list
        Set the remaining patterns as the input document
    return data
```

Fig. 4.Algorithm for Data Extraction

C. Structuring the data

The World Wide Web is a vast and rapidly growing source of information. Most of this information is in the form of unstructured text, making the information hard to query. There are, however, many web sites that have large collections of pages containing structured data, that is data having a structure or a schema. These pages are typically generated dynamically from an underlying structured source like a relational database. An example of such a collection is the set of book pages. The data in each book page has the same schema, that is, each page contains the title, list of authors, price of a book and so on.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

The algorithm automatically extracting structured data encoded in a given collection of pages, without any human input like manually generated rules or training sets. To automate the wrapper generation and the data extraction process, the proposed system develops a novel technique to compare XHTML pages and generate a wrapper based on their similarities and differences. After learning the extraction rule from a set of web documents the system print the data in a structured format.

IV. Performance Evaluation

The performance of automated data extractor is done on a collection of web documents from 50 datasets. The proposed system is an unsupervised technique, that is they learn a rule that extracts as much data as possible, give each capturing group a computer-generated label, and it is the responsibility of the user to assign a meaning to these labels. Since this system handcrafted annotations for every web document in the datasets, find which extracted data group was the closest to each annotation. To do so compared each piece of text extracted to every annotation, and computed the number of true positives (tp) and false positives (fp), since this allowed us to plot ROC curve. True positive rate, or the recall measures the proportion of positives that are correctly identified and false positive measures incorrectly identified data.

A receiver operating characteristic (ROC), or ROC curve is a graphical plot that is used to illustrate the performance of a data extractor with various datasets. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The graph in Fig.5 plots ROC curves which show excellent accuracy. Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test.

The conclusion is that, this proposed technique can achieve the maximum area under ROC curve, which means that they can extract information with perfect precision and recall in most cases.

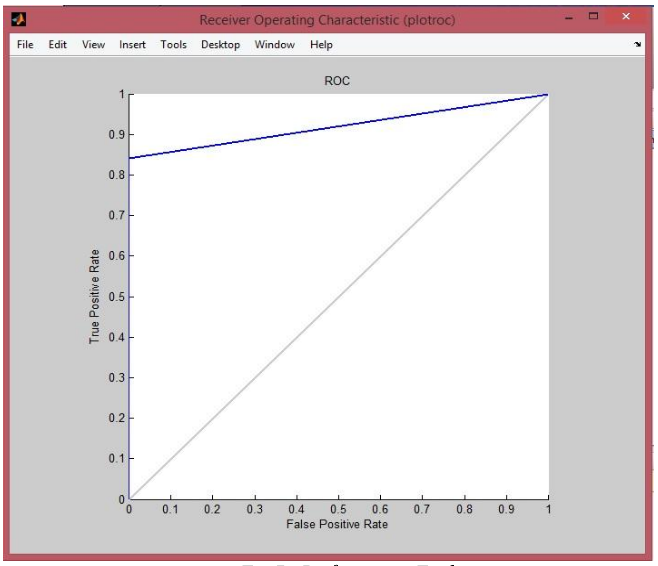


Fig.5. Performance Evaluation

V. Results

Web data extraction involves the process of extracting relevant data from XHTML encoded web pages. Thus data extraction is taking an unstructured form of data and parsing that information into a structured data set. Fig.1. shows the general view of the proposal. The proposed data extraction algorithm is implemented with python. The input of the system is a collection of web documents with same template. This built on the hypothesis that shared pattern not provide any relevant data and are part of template. Using the algorithm in fig.2 find the shared patterns and these shared patterns are organized into a list and later parsed it with capturing groups that represents the template that was used to generate the input documents. Finally by using this regular expression we search for the shared patterns in the documents and remove it from the template and then find the unshared chunk from the input document which is the relevant data. Fig.3. illustrates the flowchart of data extraction algorithm.

VI. CONCLUSION AND FUTURE WORK

Web documents are getting more and more worldly, which complicates the task of information extraction processes. Web data extraction has been an important part for many Web data analysis applications. Web data extraction involves the process of extracting web pages of text-based markup language such as HTML and XHTML which usually contain relevant information. Proposed system is an unsupervised technique which deduces the relevant data from XHTML encoded Websites. Proposed algorithm takes a set of template-generated pages as input and deduces the unknown template used to generate the pages and the values encoded in the pages are extracts as output. This built on the hypothesis that shared pattern are not likely to provide any relevant data and are part of template. Proposed system extract relevant data from web documents without any major computational impacts on the system. Performance evaluation on a large number of input page collections indicates that the algorithm correctly extracts data.

REFERENCES

1. Hassan A. Sleiman and Rafael Corchuelo Trinity: On Using Trinary Trees for Unsupervised Web Data Extraction "IEEE Trans. Knowl. Data Eng., VOL. 26, NO. 6, JUNE 2014.
2. V. Crescenzi, G. Mecca, and P. Merialdo, “Road runner: Towards automatic data extraction from large web sites," in Proc. 27th IEEE Int. Conf. Rome, Italy, pp. 109–118, 2001.
3. A. Arasu and H. Garcia-Molina, “Extracting structured data from web pages," in Proc. 2003 ACM SIGMOD, San Diego, CA, USA, pp 337-348.
4. M. Kayed and C.-H. Chang, “FiVaTech: Page-level web data extraction from template pages,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 2, pp. 249–263, Feb. 2010.
5. W. Liu, X. Meng, and W. Meng, “ViDE: A vision-based approach for deep web data extraction,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 3, pp. 447–460, Mar. 2010.
6. H. A. Sleiman and R. Corchuelo, “TEXT: An efficient and effective unsupervised web information extractor" IEEE Knowl.Based Syst., vol. 39, pp. 109-123, Feb. 2013.
7. Xiaoqing Zheng, Yiling Gu, and Yinsheng Li, “Data Extraction from Web Pages Based on Structural-Semantic Entropy "ACM, April 16-20, 2012
8. J. Wang and F.H. Lochovsky, “Data-Rich Section Extraction from HTML Pages,"Proc. of IEEE Trans. on Web Information Systems Eng. (WISE), pp. 313-322, 2002
9. H. A. Sleiman and R. Corchuelo, “A survey on region extractors from web documents" IEEE Trans. Knowl. Data Eng., vol. 25, no. 9, pp. 1960–1981 Sept. 2012.
10. Wachirawut Thamviset, Sartra Wongthanavasu “Bottom-Up Region Extractor for Semi-Structured Web Pages” in ProC. of the 2014 IEEE Int. CSE Conf., pp 284-289, 2014
11. Y. K. Shen and D. R. Karger, “U-REST: an unsupervised record extraction system," in Proc. of the 16th Int. Conf on World Wide Web, pp 1347-1349,2007.
12. H. A. Sleiman and R. Corchuelo, “Unsupervised Extraction of Product Information from Semi-structured Sources" in Proc. 13th Int. Conf. WISE,Paphos, Cyprus, pp 1544-1555, 2012.

BIOGRAPHY

Roselin Paul is a M.Tech student in Computer Science and Engineering under the discipline of Department Of Computer Science and Engineering, Viswajyothi College Of Engineering And Technology, Vazhakulam, Ernakulam Dist., Kerala, India. She received Bachelor of Engineering in Information Technology (BTech) degree in 2013 from MG University college of Engineering Thodupuzha, Idukki dist, Kerala, India. (E-mail: roselinpaul91@gmail.com).