A Hybrid Unsupervised Web Data Extraction using Trinity and NLP

Anju R  
Department of Computer Science and Engineering  
Nehru College of Engineering and Research Centre  
Thrissur, India

Mary Mareena P V  
Department of Computer Science and Engineering  
Nehru College of Engineering and Research Centre  
Thrissur, India

Abstract

Web is a huge repository of data. In order to automatically extract relevant data from web documents, web data extractors are used. The proposed technique works on two web documents that are generated by the same server-side template and learns a regular expression which represents the template of the web document. The regular expression generated can be later used to extract data from other similar documents. The proposed technique builds on the hypothesis that template introduces some shared pattern that do not provide any relevant data. In the regular expression the capturing groups represent the data. The semantic label for each capturing group is provided based on the POS (Part Of Speech) tagging. This technique could even work with the malformed documents. Hence the input errors do not have negative impact on the effectiveness.

Keywords: Web data extraction, Wrapper induction, Unsupervised Techniques, NLP, POS tagging

I. INTRODUCTION

The World Wide Web is a vast and rapidly growing source of information. The most HTML standard does not follow the strict rules, only XHMTL standard follows the strict rules. Hence most of the web pages are malformed. In this paper a web data extractor is created which would automatically extract relevant data from both well-formed and malformed web document.

Web data extractors mainly rely on extraction rules, which can be learned automatically using either supervised techniques [1], [2], i.e., techniques that require the user to provide samples of the data to be extracted or unsupervised techniques [3],[4],[5],[6], i.e., techniques that learn rules automatically and extract as much prospective data as they can, and the user then selects the relevant data from the results. Initially supervised techniques were developed. The main disadvantage of supervised techniques is that it requires users to provide labeled examples. To reduce human effort unsupervised techniques was developed.

This paper studies the problem of extracting relevant data from the similar malformed web documents, without any manually generated rules or training sets. In similar web documents, the template followed would be almost same, i.e., the tags would be same but attribute for same data may differ, and the data would be different. In order to extract the relevant data from web documents, first the template of the web page must be identified. An important challenge in automatically identifying the template is to differentiate between the text that is part of a template and the text that is part of data. The main idea to identify the template is to look for the shared patterns in two or more web documents. The template identified is represented using the regular expression. The generated regular expression is used to extract relevant data from similar documents.

The literature provides many proposals to create web data extractors. The proposals WIEN [1] and SoftMealy [2] are the examples of supervised techniques. The supervised techniques identify the extraction rules from the labeled training examples. The labeling of the training example is time consuming and not efficient enough. To reduce human effort unsupervised techniques were introduced. The main advantage of the unsupervised technique is that no user training examples are needed for web data extraction. RoadRunner [3], ExAlg [4], FiVaTech [5] and Trinity [6] are the examples of unsupervised technique. All the proposals except Trinity require the input document to be well-formed.

II. RELATED WORK

This paper is closely related to the proposals RoadRunner, ExAlg, FiVaTech and Trinity.

A. Roadrunner

RoadRunner [3] is the technique for automatically extracting the data from the HTML sites. It needs input documents to be well formed. In RoadRunner, data is extracted through the use of automatically generated wrappers. These wrappers are generated based on the similarity and difference between the web pages. The pattern discovery is based on the study of similarities and dissimilarities between the pages. In order to tell meaningful patterns from meaningless ones, this system works with two HTML pages at a time. The previous approaches to wrapping websites are based on the hand-coded wrappers. But manually writing these wrappers are difficult and labor intensive. To overcome this problem the concept of RoadRunner is introduced. It is a novel approach to wrapper inference for HTML pages. It uses a partial rule (which is initialized to any of the input documents) to parse...
another document and applies a number of generalization strategies to correct the partial rule when mismatches are found. The mismatches are used to identify relevant structures. The main limitation of the RoadRunner is the numbers of errors in the input documents affect the effectiveness.

**B. ExAlg**

ExAlg [4] is an approach for extracting the structured data from a collection of web pages generated from the common template. ExAlg consist of two stages such as Equivalence Class Generation Stage (ECGM) and Analysis stage. In ECGM stage, find the sets of tokens having the same frequency of occurrence in every page which are known as equivalence classes. There may be many equivalence classes. ExAlg retains only the equivalence classes that are large and whose tokens occur in a large number of input pages, such type of equivalence classes are known as LFEQs (for Large and Frequently occurring EQivalence classes). In analysis stage, the LFEQs are refined by discarding invalid ones and learn a regular expression. The problem identified in the ExAlg is that it is not clear ExAlg whether ExAlg can work on malformed input document or not.

**C. FiVaTech**

FiVaTech [5] is an unsupervised page-level web data extraction technique. This technique automatically detects the schema of a website. The FiVaTech introduces a new tree structure, called fixed/variant pattern tree. This tree structure is used to identify the template and detect the data schema. FiVaTech contains two stages. The first stage is merging input DOM trees to construct the fixed/variant pattern tree. The second stage detects structure of the website, i.e., to identify the schema and define the template. To find such a common template requires multiple pages or a single page which contains multiple records as input. The limitations of the FiVaTech are searching the longest repeating patterns is a time consuming process and also it does not work on the malformed input document. In order to work with the malformed input document it is needed to be corrected, which have a negative impact on its effectiveness.

**D. Trinity**

Trinity [6] is an unsupervised web data extraction technique that learns extraction rules from a set of similar web documents. It can take both well and malformed documents as input. Previous techniques require the input documents to be well-formed otherwise it requires the input documents to be repaired beforehand. Trinity takes two or more similar web documents as input and searches for a shared pattern using KMP algorithm. Whenever a shared pattern is found, it partitions the input documents into prefixes, separators and suffixes. The prefixes are the fragments from the beginning of the input document to the first occurrence of a shared pattern. The separators are the fragments in between successive occurrences. The suffixes are the fragments from the last occurrence until the end of input document. These fragments are analyzed recursively until no shared patterns are found. These prefixes, separators and suffixes are organized into a trinary tree that is later traversed to build a regular expression. The expression represents the template which can be used to extract data from other similar documents. In trinity, since the malformed document does not have to be repaired the input errors do not have negative impact on its effectiveness.

### III. System Overview

The proposed system creates an unsupervised web data extractor that learns extraction rules from a set of similar web documents. It can extract data from both well and malformed documents. Fig. 1 shows the dataflow diagram of proposed system.

![Fig. 1: Data Flow Diagram of Proposed System](image)

There are mainly two phases in the proposed system. They are Regular Expression Generation and Extraction. In the first phase, the proposed system first takes two similar web documents as input, as in Trinity, and tokenize each document separately. In Trinity, whenever a shared pattern is found, it partitions the input documents into prefixes, separators and suffixes and organizes them into trinary tree. The proposed technique identifies shared pattern by token comparison and stores the shared pattern. The shared pattern is a sequence of tokens that are repeating in all web documents. The text after the shared pattern would be most probably data. In similar documents the text in between the repeating shared pattern would be relevant data. The relevant data would occur in similar order. The shared patterns identified are used to construct the regular expression. The regular expression generated is used to extract data from similar documents.

In the second phase, a similar web document is taken as input. By using the regular expression generated in the previous phase the data is extracted from the web document. The regular expression represents the template of a web document. After data extraction, the label for each data is extracted using the POS tagging. The data and label extracted is stored in a XML document.
IV. REGULAR EXPRESSION GENERATION

Fig. 2 shows the data flow diagram of Regular Expression Generation phase. In this phase, first two web documents are selected as input. Each document is tokenized separately. Then the tokens of both documents are compared with each other and identify the shared pattern. After finding all the shared patterns, then the regular expression is generated from them. The regular expression represents the template of a web document. This regular expression is used in the next phase to extract data from similar documents. Regular expression is a way to describe a set of strings based on common characteristics shared by each string in the set. They are used to search, edit, or manipulate text and data.

A. Tokenization
In Fig. 2, first two web documents are taken as input which follows the complete template that could occur in all similar documents. Then each document is tokenized separately. Here, each document is considered as a string. Then the string is broken into tokens using whitespace character as delimiter. A whitespace character can be space character, tab character, carriage return character, new line character, vertical tab character or form feed character. When a web document is tokenized, each tag and each word of the text is obtained as tokens.

B. Identifying Shared Patterns
The proposed technique identifies shared pattern by token comparison. The shared pattern is a sequence of tokens that are repeating in all web documents. The tokens of both documents are compared with each other. When a token match is found, that token would be the starting token of the shared pattern. The tokens till a mismatch are found form a shared pattern. The sequence of tokens from the starting token to the token at the position where the last match is found forms a shared pattern. In the same manner, all the shared patterns is found by token comparison. After finding all the shared patterns, then the regular expression is generated from them.

In Trinity, the shared pattern is found by character by character comparison. In the proposed technique, the shared patterns are found by the html tag comparison which takes advantage of the fact that the html tags in similar documents would be similar.

C. Generating Regular Expression from Shared Patterns
The shared pattern stored is combined in the order in which it is found. Whenever the repeating shared pattern is found, it is placed in between the parenthesis and the plus closure is added at the end. In this manner the regular expression is constructed from the shared patterns. The set of same kind of data is called a capturing group, such as capturing group of author names. In the regular expression, these capturing groups are used to capture the particular kind of data. The regular expression is generated in such a way that the data can be extracted from the similar document without any tokenization required.

V. EXTRACTION

Fig. 3 shows the data flow diagram of the Extraction phase. This phase includes the extraction of both data and label for each data. The data is extracted by using the regular expression generated in the first phase. For label extraction, first POS tagging is
applied on the web document. After POS tagging, a POS tag is attached to each text in the web document. Using the extracted data and the POS tags in the web document the label for each data is extracted from the web document itself.

A Hybrid Unsupervised Web Data Extraction using Trinity and NLP

Data Extraction

A regular expression is an expression which represents a set of strings. Here the set of strings are data to be extracted. In a regular expression, the data is represented by capturing groups. The data in web document is identified by searching for an open parenthesis. If it ends by a close parenthesis only, then the matching text indicates data and the text is extracted. If the parenthesis ends by close parenthesis followed by a plus sign, it indicates that the pattern is repetitive, so there could be many matches and extracts each data separately.

POS Tagging

POS tagging is an important tool for processing natural languages. It is a process of assigning appropriate parts of speech tags like noun, verb, adjective and adverb to each word in an input sentence. POS tags are also known lexical tags to choose correct grammatical tag for word on the basis of linguistic feature. When POS tagging is applied to a web document, it would consider the html code of the web document as an English text and assign a POS tags to each word in the document.

Label Extraction

In Trinity, user provides the label for each capturing groups. In the proposed technique, the label for each capturing group provided using POS (Part Of Speech) tagging. Sometimes in html document a label will be provided for each data. In similar web documents, the label provided for each data would be same. After POS tagging, the tag associated with each label would be proper noun, as is shown in the example above. By searching for words with proper noun tag, the label associated with each data could be extracted from the html document itself.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a new approach to create an unsupervised web data extractor. It is based on the hypothesis that the web documents generated by the server side template provides shared pattern which does not provide any relevant data. It can extract data from both well and malformed documents. Hence the input errors do have negative impact on its effectiveness. The main challenge of this technique is to identify that the web documents for learning the template because they must follow the exact template. In this technique the data extracted is short words. But the web documents may have unstructured text which may contain the relevant data. The relevant data from unstructured text can be extracted by NLP (Natural Language Processing). Also, the web documents may contain user comments. It is possible to identify if a user comment is positive or negative by sentiment analysis.

REFERENCES


