International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 4 2239 – 2243

Context-free Grammar Extraction form Web Document using Probabilities Association

Ramesh Thakur
International Institute of Professional Studies
Devi Ahilya University
Indore, India
r_thakur@rediffmail.com

Abstract—The explosive growth of World Wide Web resulted in the largest Knowledge base ever developed and made available to the public. These documents are typically formatted for human viewing (HTML) and vary widely from document to document. So we can't construct a global schema, discovery of rules from it is complex and tedious process. Most of the existing system uses hand coded wrappers to extract information, which is monotonous and time consuming. Learning grammatical information from given set of Web pages (HTML) has attracted lots of attention in the past decades. In this paper I proposed a method of learning Context-free grammar rules from HTML documents using probabilities association of HTML tags.

Keywords- *Knoledge discovery, Grammatical inference, Context-free grammar.*

*****

I. INTRODUCTION

The Explosive growth of the World-Wide-Web has resulted in huge amount of information source on the Internet. Generally these information are semi-structured (HTML), we can also find structured and unstructured. The information is also dynamic, it contains hyperlinks and may be represented in different forms and is globally shared over multiple sites and platforms. The web is driving force of research on information extraction form semi-structured data.

There is different view about Web Pages. Some researchers define all Web Pages as semi-structured information. As they all contain the structuring information concerning display style *i.e.* HTML tags. These tags are the instruction to browser for presentation. However [1] give a better categorization of types of web pages. A Web Page that provides itemized information is structured, if each attribute in a tuple can correctly be extracted based on some uniform syntactic clues, such as delimiters or the orders of attributes. Semi-structured Web Pages, however may contains tuples with missing attributes, attributes with multiple values, variants attribute permutations, and exceptions. A Web Page is unstructured if linguistic knowledge is required to extract the attributes correctly. Since the semantic of Web Page are restricted to each Web page or a class of Web pages, hence it is not fully structured information. We consider Web Pages are semi-structured information.

Information Extraction (IE) is different from the more mature technology of Information Retrieval (IR). Rather than to extract information the objective of IR is to select a relevant subset of documents from a large collection based on user query. Manning and Raghavan [2] described Information Retrieval (IR) as follows: "Information retrieval (IR) is to find the material (usually documents) of an unstructured nature (usually text) that satisfies information need from within large collections (usually stored on computers)". In Contrast, goal of Information extraction (IE) is to extract relevant information from the documents. Hence the two techniques are complementary, and used in combination they can provide more powerful tools for text processing [3].

Not only the IE and IR differ in aims, they also usually differ in the technique. The IE has emerged from research on rule-based system in computational linguistic and natural language processing, while information theory, probability theory and statistics have influenced the IR [3].

The learning of the syntax of the language is usually referred to as grammatical inference or grammar induction. The product of this process is a grammar, a formalism that captures the syntax of a language. The objective of the grammatical inference to infer a formal language, such as context-free grammar, which describes the given sample set. These grammar rules will be used to create structural descriptions of the unstructured and semi-structured documents. In automated grammar learning, the task is to infer grammar rules from given information about the target language. Information Extraction from textual data has various applications, such as semantic search [4].

If the sentences confirm to a language described by a known grammar, several techniques exist to generate the syntactic structure of these sentences. Parsing [5] [6] is one of such technique that rely on knowledge of grammar.

A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience [7]. When we use this general definition we can say that the task is to learn a grammar, the performance measure could be a metric that calculates the difference between the grammar found and the target grammar (*i.e.*, the grammar to be learned) and the experience could be the linguistic input in one or another form (*e.g.* unstructured or semi-structured text).

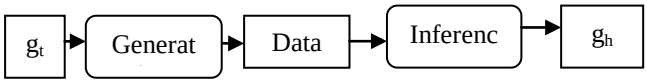

Figure 1. :Induction process for the target grammar [8].

II. CONTEXT-FREE GRAMMAR INFERENCE

The problem of learning the correct grammar for the unknown language from finite example is known as grammatical inference problem. Context-free grammar is a powerful and convenient formalism for representing document structures. We therefore obtain a common structural model for

 ISSN: 2321-8169
2239 – 2243

documents as a problem in grammatical inference. For example consider a web page for which we want to learn the grammar [9].

DAVV

Institute of Engineering and Technology

International Institute of Professional Studies

School of Computer Science

Figure 2. Sample Web page.

```
<html>
<body>
<h1> DAVV</h1>
<p> <img src="logo.gif">Institute of Engineering and Technology</p>
<p>International Institute of Professional Studies</p>
<p>School of Computer Science</p>
</body>
</html>
```

Figure 3. HTML code for sample web page.

To make above HTML code simpler first we convert this to abstract strings by using the tokens html tags, and text to represent the running text.

```
<html>
<body>
<h1> text</h1>
<p><img src="image.gif">text</p>
<p>text</p> <p>text</p>
</body>
</html>
```

Figure 4. Resulting tokens of sample web page.

The grammatical inference results in following grammar.

$S \rightarrow$ *<html><body>XY</body></html>*

$X \rightarrow$ *<h1>text</h1>*

$Y \rightarrow$ * Z text Y* | ϵ

$Z \rightarrow$ *<img text>* | ϵ

Where the start symbol *S* represents a complete page, the non-terminal *X* represents a header, and the non-terminal *Y* represents a paragraph item. Since *Y* recursively contains itself as a production, this grammar permits an arbitrary number of repetitions of *Y*. Using the above grammar we have generated following set of string.

<html><body>< h1> text</h1></body></html>

<html><body> < h1> text </h1>< p> text </body></html>

<html> <body>< h1> text </h1> text text </body>< /html>

The above grammar generates a language that syntactically appears to be a generalization of the input set and its productions also model the semantic structure of the language.

III. STOCHASTIC CFG

A CFG G is stochastic CFG (SCFG) that includes an assignment of weights, with each weight being between zero and one (inclusive), to the production, such that, for every non-terminal A, the sum of weights of all the A-production in *G* is 1.

$$\sum_{\alpha} P(A \rightarrow \alpha) = 1$$

The assigned weights are also called as probabilities, and $A \rightarrow \alpha$ (p_r) denotes the assignment of weight pr to the production $A \rightarrow \alpha$. The weight of derivation

$A \Rightarrow \alpha_1 \Rightarrow \alpha_2 \ldots\ldots \Rightarrow \alpha_n$ is defined recursively as:

- $P(A)=1$
- $P(A \Rightarrow \alpha_1 \Rightarrow \alpha_2 \ldots\ldots \Rightarrow \alpha_{n-1} \Rightarrow \alpha_n) = P(A \Rightarrow \alpha_1 \Rightarrow \alpha_2 \ldots\ldots \Rightarrow \alpha_{n-1})\ P(B \rightarrow \gamma)$

Where $B \rightarrow \gamma$ is the production used in the step $\alpha_{n-1} \Rightarrow \alpha_n$

The probability of deriving a string *w* is denoted by the sum:

$$P(w) = \sum_{A \Rightarrow \cdots \Rightarrow w} P(A \Rightarrow \cdots \Rightarrow w)$$

The grammar *G* is consistent if it defines a well-formed weight distribution over all strings [10].

$$\sum_{w \in \Sigma^*} P(w) = 1$$

IV. GRAMMATICAL INFERENCE ALGORITHM

The algorithm considers the relative complexity of candidate grammar. For convenience we take the hypothesis to the set of stochastic context-free grammars. Stochastic context-free grammar is the context-free grammar with probabilities attached to their productions. This augmented space is more continuous than the space of standard context-free grammars and provide more freedom for modifying candidate grammars. For example, the production $X \rightarrow u$ can be continuously deformed into the production $X \rightarrow u \mid w$ by varying the probabilities. The probabilities also make it possible to quantitatively assess the fit between the candidate grammar and language sample by calculating the probability that the given grammar would have generated.

We split the problem of grammatical inference into following phases:

ISSN: 2321-8169
2239 – 2243

- Codification of string: first the algorithm we transformed the data into suitable format. The algorithm expects a set of positive sequence of symbols from a finite alphabet set. So the strings (sentences) of input data sets are codified based on their syntactic categories.
- Calculate probabilities: for all sub string calculate the probabilities $w_1 \mid w_2 \mid w_3 \ldots \mid w_n$, $[p_1, p_2, p_3, \ldots\ldots, p_n]$ where $w_1, w_2, w_3, \ldots w_n$ are string occurring in the sample and $p_1, p_2, p_3, \ldots p_n$ are their relative frequencies. If all string are different, then the p_i will be equal to 1/m where m is number of string how ever the p_i may very if some string appear more than once in the sample.
- Discovery of pattern: Searching of repeated sub-string *s* are performed and the sub-string *s* occur multiple times indicated by its associated probabilities a new grammar $Y \rightarrow s$ are added and all occurrences of *s* are replaced by are replaced by are replaced by *Y*.
- Multiple Production alternative: If the occurrence of *s* is in such a position that multiple production alternatives are possible ($X \rightarrow us \mid ws$) then new production is $Y \rightarrow u \mid w$ and $X \rightarrow Ys$
- Redundant production: Merge redundant rules and drop production, which are inaccessible (cannot be reached from start symbol).

Proposed Grammatical Inference by Relative Probabilit) Algorithm

Input: A corpora C of flat sentence (HTML Codified string).

Max_number(substring)

Output: Set of CFG rules R

Begin

Initialize rule set *null*
Calculate sub-sub-strings
for every sub_string having length >1 do
calculate the relative Probabilityof sub-string β
calculate_relative_probability of sub string
select *sub_String* of highest Probabilities
select next non terminal symbol for LHS of CFG
add new rule to rule $N \rightarrow \gamma$ to set *R*

else
if γ has multiple production alternative resolve it.

end for
end.

Procedure calculate_relative_probability(*u*)
// This procedure return relative Probability of HTML TAGS for the sub string *u* in the corpora.
Begin
relative_Probability=0.0
m=number_substring(Σ)
for $\forall \alpha \in C, \mid \alpha \mid > 1$ do
relative_friquency =(count_of_substring_u_in_α)/*m*
end for
return(relative_Probabilities)
end

Figure 5. Grammatical inference Proposed algorithm.

V. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented using a code written in *C* programming language. The main data structures are stored in array of character and for sub-string and relative frequency count the array of structure (string and double field) are used. It uses the probabilistic relative frequency for replacement rule in the corpora.

```
<html><head><title>Listing of NOKIA Mobile Phone</title></head>
<body>
<h1>India Price</h1>
<table border=1 width=100%>
<tr><td><b><ahref="nokia-7_1146.html">Nokia E7
</a></b><br>Rs.25691<br> A QWERTY plus touchscreen business phone
running on Symbian^3operating system.
</td></tr><tr><td><b<a href="nokia-n900_897.html">Nokia
N900</a></b><br>Rs.23529<br>
Nokia N900 is a high performance mobile computer from Nokia having the
latest Maemo operating system in it. Its Linux-based Maemo software takes us
into a new era of mobile computing. Has a powerful processor, large internal
storage and a sharp touch screen display.</td></tr><tr><td><b> <a
href="nokia-n97-mini_898.html">Nokia N97 Mini</a>
</b><br>Rs.17788<br> 12 MP Camera Phone with AMOLED Touchscreen
and HD Video Recording</td></tr>
<tr><td><b> <a href="nokia-x6-16gb_1080.html">Nokia X6 16GB</a></b>
<br>Rs. 330,00<br>Auto Exposure, Auto Focus, Carl Zeiss Optics, Exposure
Compensation, Flash, Full Screen Viewfinder, Geotagging, Self Timer,
Sequence Mode, Video Light, Video Stabilization </td></tr>
</table>
</body>
</html>
```

Figure 6. A part of HTML code for Mobile listing.

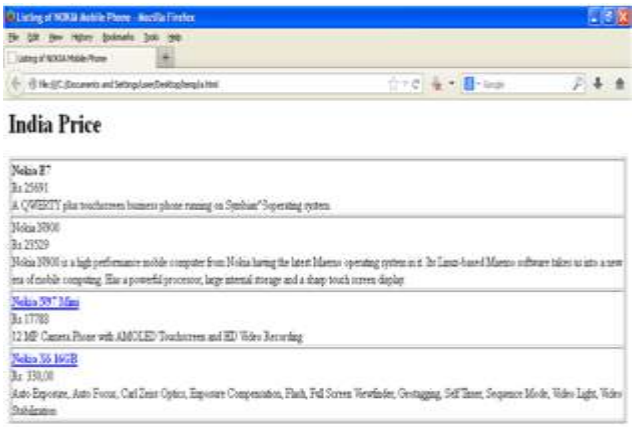

Figure 7. Sample Mobile Web page.

A. *Codification of HTML String*

Proposed algorithm has been implemented using a code written in *C* programming language. The main data structures are stored in array of character and for sub-string and relative frequency count the array of structure (string and double field) are used. It uses the probabilistic relative frequency for replacement rule in the corpora.

 ISSN: 2321-8169
2239 – 2243

```
<html><head><title>text </title></head>
<body>
<h1>text</h1>
<table text%>
<tr><td><b><a
href=text>text</a></b><br>text<br>text</td></tr>
<tr><td><b><a
href=text>text</a></b><br>text<br>text</td></tr>
<tr><td><b><a href=text>text</a>
</b><br>text<br>text</td></tr>
<tr><td><b><a href=text>text</a></b> <br>text<br>text
</td></tr>
</table>
</body></html>
```

Figure 8. HTML code of Sample web page.

B. *Iterations Steps for Algorithm.*

```
Z→<a href=text>test</a> | e
<html><head><title>text </title></head>
<body>
<h1>text</h1>
<table text>
<tr><td><b>Z</b><br>text<br>text</td></tr>
<tr><td><b>Z</b><br>text<br>text</td></tr>
<tr><td><b>Z </b><br>text<br>text</td></tr>
<tr><td><b>Z</b><br>text<br>text /td></tr>
</table>
</body>
</html>
Y→<tr><td><b>Z</b>
<br>text<br>text</td></tr>

<html><head><title>text </title></head>
<body>
<h1>text</h1>
<table text>
        YYYY
</table>
</body>
</html>
```

……

Figure 9. Intermediate results of proposed algorithm.

C. *Results (Grammar Inference)*

After applying the proposed algorithm the following grammar results were inferred:

X→<html><head><title>text
</title></head><body><h1>
text</h1><table text> YYYY</table></body></html>
*Y→<tr><td>Z
text
text</td></tr>*
Z→test |e

We can interpret this as follows: the start symbol *X* represents a complete page. We can see that a page begins with fixed preamble followed by a variable number of occurrences of *Y* each represents a single listing. A listing consists of a table raw, which contains reference to their site *Z*. The non-terminal *Y* corresponding to a listing we may generate a wrapper that segments each listing by searching the pattern specified by *Y*. The data fields for each listing can be extracted by mapping the text symbols to their actual content. Then the domain specific heuristics can be used to identify the semantic meaning of different fields. Science domain specific knowledge is not used in grammar generation it is used in the last step so this approach can be easily used in other domain. After applying the procedure we get attribute value.

VI. Evaluation of Algorithm

The evaluation of Information Extraction using grammatical inference problem has different approaches. Generally, the evaluation of grammar inference algorithm is carried out by giving input to the algorithm a set of unstructured data and evaluating its output (grammar rules). Three principal evaluation strategies usually applied for evaluating grammar inference algorithm [11].

- Looks-Good-to-me,
- Compare Against Treebank,
- Rebuilding Known Grammars.

The *Rebuilding Known Grammars* approach is another evaluation strategy. This method, starting from a pre-defined (simple) grammar, generates a set of example sentences, which are given as input to the grammar inference algorithm and the resulting grammar is compared manually to the original grammar. If the inferred grammar is similar or equal to the original grammar then the learning system is considered good.

We have used the *Rebuilding Known Grammars* evaluation strategy for the evaluation of our proposed algorithms. The following metrics have been used to compare the grammar learned by the proposed algorithms.

Precision, which measures the number of correctly learned constituents as a percentage of the number of all learned constituents. The higher the precision, the better the algorithm is at ensuring that what has been learned is correct.

$$\text{Precision} = \frac{\sum \text{Correctly Learned Constituentes}}{\sum \text{Learned Constituentes}}$$

Recall, which measures the number of correctly learned constituents as a percentage of the total number of correct constituents. The higher the recall, the better the algorithm is at not missing correct constituents.

$$\text{Recall} = \frac{\sum \text{Correctly Learned Constituentes}}{\sum \text{possible correct Constituentes}}$$

When comparing the performance of different systems, both precision and recall must be considered. However, as it is not straightforward to compare the two parameters at the same time, various combination methods have been proposed. One such measure is *F-Score*, which combines precision, *P* and recall *R*, in a single measurement as follows:

$$F = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Using the F-score, the relative performance of systems reporting different values for recall and precision, can easily be compared.

TABLE I. RESULTS OF PROPOSED ALGORITHM.

Data Set	Proposed Proposed Algorithm			
	Corpus size (Sentences)	Precision %	Recall %	F-Score %
Sample set one	986	78.6	79.1	79.8
Sample set two	878	64.6	44.2	52.5
Average	---	**71.6**	**61.65**	**66.1**

It is clearly observed that both the precision and recall of proposed system are found higher. Also after averaging the Precision, Recall and F-score values, we found that the proposed algorithm have satisfactory results.

VII. CONCLUSION

We have proposed a general approach for generating information extraction wrappers using grammatical inference that enables information extraction from the semi-structured document (HTML). It does not require the manually labeling of example for data incentive sites. In this work we have extracted attribute value of frequently occurring data from data intensive sites. This work can be seen as a basic component of the larger goal of extracting knowledge repositories from the web.

REFERENCES

[1] C-H Hsu and M-T Dung "Generating Finite-State Transducers for semi-structured Data Extraction From the Web". *Information System*, Vol 20. No. 8, pp 521-538, 1998.

[2] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "An Introduction to Information Retrieval", *Cambridge University Press Cambridge, England Online edition Cambridge UP*, 2009..

[3] R. Gaizauskas, Y Wilks. "Information Extraction: Beyond Document Retrival", *Computational Linguistics and Chinese Language Processing*, vol. 3, no. 2, pp 17-60, 1998.

[4] P. Palaga, L. Nguyen, U. Leser, and J. Hakenberg, "High-performance information extraction with AliBaba ," *In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '09 ACM New York* pp 1140–1143, 2009.

[5] Allen J., "Natural Language Understanding," *The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, USA. Second Edition*, 1995.

[6] N. A. Chinchor, Overview of MUC-7/MET-2 1998.

[7] Mitchell, T. M. "Machine learning" New-York, USA: MIT Press, McGraw-Hill pp 2–3, 1997.

[8] Menno M. van Zaanen "Bootstrapping Structure into Language Alignment-Based Learning*", Phd thesis, The University of Leeds School of Computing*, 2001.

[9] Ramesh Thakur, Suresh Jain, Narendra S. Chaudhari, Rahul Singhai " Information Extraction from the Un-Structured Document using Grammatical Inference and Alignment Similarity". *In Proceedings 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) Sciencedirect Procedia Technology 4 (2012) Published by Elsevier Ltd. ISSN 2212-0173*, pp 365 – 369, 2012.D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," Science, vol. 294, Dec. 2001, pp. 2127-2130, .

[10] Stolcke, Andreas Bayesian. "Learning of Probabilistic Language Models". *PhD thesis, University of California at Berkeley*, 1994.

[11] D'Ulizia, Arianna, Fernando Ferri, and Patrizia Grifoni. "A survey of grammatical inference methods for natural language learning." *Artificial Intelligence Review* 36, No. 1 pp 1-27, 2011.