International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 4 2269 – 2272

Segmentation of Document Using Discriminative Context-free Grammar Inference and Alignment Similarities

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

Abstract—Text Documents present a great challenge to the field of document recognition. Automatic segmentation and layout analysis of documents is used for interpretation and machine translation of documents. Document such as research papers, address book, news *etc*. is available in the form of un-structured format. Extracting relevant Knowledge from this document has been recognized as promising task. Extracting interesting rules form it is complex and tedious process. Conditional random fields (CRFs) utilizing contextual information, hand-coded wrappers to label the text (such as Name, Phone number and Address *etc*). In this paper we propose a novel approach to infer grammar rules using alignment similarity and discriminative context-free grammar. It helps in extracting desired information from the document.

Keywords-*Labeling document, Information Extraction, Alignment Profiling.*

*****

I. INTRODUCTION

Search engines helps reader in finding document of their interest based on keyword searching. The coverage and precision of search engine are of significant importance. This quality presently depends on information retrieval component that extracts meta-data from page header and references. These meta-data are furthered used in many component applications such as field-based search, etc. There are many application require similar approach such as address book, resume, news *etc*. In automated grammar learning, the task is to infer grammar rules from given information about the target language. The sentences are given as examples for such learning.

It is widely acknowledge that the principle goal in linguistics is to characterize the set of sentence-meaning pairs. The linguistics deals with production and perception of language, using sentence-meaning pairs corresponds to converting from sentence to meaning in perception and from meaning to sentence in production. Directly finding these sentence-meaning is difficult. A system first analyses the sentence, and generate the intermediate structure, and using these structure the mining of sentence is computed [1]. Based on the principle of compositionality of meaning [2], the meaning of a sentence can be computed by combining the mining of its constituents. The constituents on position *X* may be more than one word.

One main approach for learning some subclasses of regular language is by splitting the merging the states in the deterministic finite automata (DFA) [3]. Prefix tree acceptors are often constructed from the given sample as a starting DFA, and they are useful for modeling positive samples. Other approaches include learning by queries [4] learning by structural information [5] learning subclass of language [6], learning by genetic algorithm[7], neural networks [8] and Markov approaches [9] other related work can be found in [10,11,12,13].

The hierarchical structure of documents is grouped into different headings fallowed by text. The form of this information varies from document to document. For effective search require field extraction also known as segmentation of document. One way of doing this is to consider the text block as a sequence of words or tokens and assign labels to each of these tokens. All the tokens correspond to particular labels. In the following example classification algorithm can be used to perform schematization (Preface, Introduction, *etc*).

We propose a grammar learning methodology for Segmentation of documents to automate the construction of context-free grammar rules and facilitate the process of Segmentation. We used discriminative grammatical inference and alignment similarities algorithm for classification and generate context-free grammar rules from documents (text document).

II. SUPERVISED AND UNSUPERVISED LEARNING

Language learning algorithms are roughly divided into two groups based on the amount of information about the language they use.

- Supervised Learning
- Unsupervised learning

All learning algorithms use a teacher that gives examples of (unstructured, semi-structured) sentences in the language. In addition, some algorithms also use a critic. A critic may be asked if a certain sentence (structure) is a valid sentence in the language or not. The algorithm uses a critic to validate hypotheses about the language. Supervised language learning methods use a teacher and a critic, whereas the unsupervised methods only use a teacher [14].

Supervised language learning methods typically generate better results. These methods receive knowledge of the structure of the language (by a critic). Unsupervised language learning methods do not receive these structured sentences, so they do not know what the output should be generated. However, it is interesting to investigate unsupervised language learning methods, since the costs of preparing knowledge of structure are time consuming and expertise intensive [15].

III. DISCRIMINATIVE CONTEXT-FREE GRAMMAR

Context-free grammars (CFGs) were firstly defined by Chomsky in the mid-1950s [16]. A context-free grammar (CFG) is defined as $G=\{N, C, P, S\}$ where N is the set of non-terminals symbol. T is the set of terminal symbols, P is the set

of production rule and *S* is the initial symbol. The language *L*(*G*) is the set of terminal string *w* that have derivations from initial symbol. *L*(*G*)= {w ∈ C* | S⇒ * W}. Context-free grammar (CFG) has production rules of the form $\{R_i : N^{j_i} \to C^j\}_{i=1}^{r}$ $N^{j_i} \in N$ and $C^j \in (N \cup C)$ *i.e.* sequence of terminals and non-terminals. We associate a score $S(R_i)$ with each rule R_i. A parse tree is a tree whose leaves are labeled by terminals and interior nodes are labeled by non-terminals. Further if a node N^{j_i} is labeled as interior node, then the child nodes are the terminals or non-terminals in C^i where $R_i : N^{j_i} \to C^j$. The score of a parse tree *T* is given by $\sum R_i : N^{j_i} \to C^j \in_T S(N^{j_i} \to c_j)$. A parse tree for a sequence of $w_1 w_2, \ldots w_m$ is a parse tree whose leaves are $w_1 w_2, \ldots w_m$.

The probabilities associated with all the rules, and a given sequence of terminals $w_1 w_2, .. w_m$ our algorithm can compute the highest scoring parse tree in time $O(m^3 r)$, where *m* is relatively small.

Probabilistic Context-Free Grammar (PCFG) can be described as $S(R_i)$ to be the logarithm of probability *P*(*Ri*) associated with the rule. It the probability *P*(*Ri*) is log-linear model and N^{j_i} can be derived from the sequence $w_a w_{a+1} \ldots w_b$ (denoted as $N^{j_i} \Rightarrow w_a w_{a+1} \ldots w_b$) and *P*(*Ri*) can be written as[17]

$$\frac{1}{Z_{(\lambda(R_i)a,b,R_i)}} \exp \sum_{k=1}^{F} \lambda_k(R_i) f_k((w_a, w_{a+1} \ldots w_b, R_i))$$

$\{f_k\}_{k=1}^{f}$ is the set of features and $\lambda(R_i)$ is vector of parameter representing feature weights (possibly chosen by alignment similarity). $Z_{(\lambda(R_i)a,b,R_i)}$ is the partition function and is chosen to ensure that the probabilities add up to 1.

A probabilistic grammar defines a language, and associated probabilities with each sentences of language. The grammar only associates probability with different parses of a particular sequence of terminals. In our algorithm the probability associated with the rule $R_i : N^{j_i} \to C^j$ is given by

$$S(R_i) = \sum_{K=1}^{F} K(R_i\ f_k(w_a, w_{a+1}, \ldots w_b, a, b, R_i)$$

When applied to the sequence $w_a w_{a+1. \ldots} w_b$ the feature can depends on all the tokens, not just the sub-sequence of tokens by N^{j_i}

IV. Alignment Similiraties for Label Selection

Our algorithm first extracts feature for generation of rules. Science documents are not prepared using any global schema. These features are similar to those used by the CRF model described in [17] however in the CRF model all the feature can only relate the sequence of observation w_i, the current state s_t.

Alignment Based Learning (ABL) is based on alignment information [11]. In ABL Pair-wise alignment for each pair of the input sentences is done by finding equal and unequal parts. Pair-wise alignment is an arrangement of two sequences, which shows where the two sequences are similar and where they differ. A good alignment shows the most significant similarities, and least differences. A score is assigned to an alignment called alignment score, to measure the goodness of an alignment. Scoring scheme is usually defined on the pairing of different constituents and gap penalty for shifts in the alignments.

A sub-sentence or word group of sentence *S* is a list of words $u_{i..j}^S$ such that $S = u + v_{i..j}^S + w$ (the + is defined to be the concatenation operator on lists), where *u* and *w* are lists of words and $u_{i..j}^S$ j with $i \le j$ is a list of $j - i$ elements where for each k with $1 \le k \le j - i$: $v_{i..j}^S[k] = S[i + k]$. A sub-sentence may be empty (when $i = j$) or it may span the entire sentence (when $i = 0$ *and* $j = |S|$). *S* may be omitted if its meaning is clear from the context.

A sub-sentence or word group of sentence *S* is a list of words $u_{i..j}^S$ such that $S = u + v_{i..j}^S + w$ (the + is defined to be the concatenation operator on lists), where *u* and *w* are lists of words and $u_{i..j}^S$ j with $i \le j$ is a list of $j - i$ elements where for each k with $1 \le k \le j - i : v_{i..j}^S[k] = S[i + k]$. A sub-sentence may be empty (when $i = j$) or it may span the entire sentence (when $i = 0$ *and* $j = |S|$). *S* may be omitted if its meaning is clear from the context.

A sub-sentences *u* and *v* are substitutable for each other if

- The sentences $S_1 = t + u + w$ and $S_2 = t + v + w$ (with *t* and *w* sub-sentences) are both valid, and
- For each *k* with $1 \le k \le |u|$ it holds that $u[k] \notin v$ and for each *l* with $1 \le l \le |v|$ it holds that $v[i] \notin u$.

Note that this definition of substitutability allows for the substitution of empty sub-sentences [18]. We assume that for two sub-sentences to be substitutable, at least one of the two sub-sentences needs to be non-empty.

We are using alignment-based similarity for extracting the tokens. Alignment based learning (ABL) [10, 11, 19, 20] is based on alignment information. In ABL Pairwise alignment for each pair of the input sentences is done by finding equal parts and unequal parts. Pairwise alignment is an arrangement of two sequences, which shows where the two sequences are similar and where they differ. A good alignment shows the most significant similarities, and the least differences. Usually a score is assigned to an alignment, called an alignment score, to measure goodness of alignment. Scoring scheme is usually defined on the pairing of different constituents and gap penalty for shifts in the alignments. In alignment-based system, more gaps means less similarity. Words that are located above each other and that are unequal in the alignment are called substitution. In an alignment, if there is a substitution then two subsequences said to be aligned in the same slot.

V. EXTRACTION OF LABELED

One of the ways of generating CFG is to use a corpus annotated with tree structure, such as the penn treebank [21]. Given such corpus, our algorithms based on counting which is used to determine the probabilities for the model. Annotating the corpora with the tree structure is done manually which is time consuming and expensive in human effort. We automatically generate the parse tree from label sequences certain class of grammar.

Given a parse tree T for a sequence $w_1\ w_2 \ldots w_m$. let the reduced parse tree T' be tree obtained by deleting all the leaves of the tree The reduced parse tree, the label sequence $l_1, l_2, \ldots l_m$ corresponds the leaves. The reduced parse tree is the tree of sequence $l_1, l_2 \ldots l_m$ over a different grammar in which the labels are the terminals. The grammar can easily be obtained from the original grammar by discarding all rules in which a label occurred on the LHS. If G' is reduced grammar then we can use g' to parse any sequence of labels. The G' can parse a sequence $l_1, l_2, \ldots l_m$ if and only if there is a sequence of words $w_1, w_2, \ldots w_m$ with l_i begins the label of w_i. we say that G is label if G' is unambiguous. To generate a parse tree for a label unambiguous grammar, for given label we use

- Generate a parse tree for label sequence using the reduced grammar G'.
- Glue on the edges of the from $l_1 \rightarrow w_i$ to the leaves of the reduced tree.

The above method gives us the unique parse tree for given sequence of words w1, w2, …. wm and their corresponding labels l1, l2, ….lm. Thus the method allows us to generate a collection of pares trees giving a collection labeled sequence.

VI. PROPOSED ALGORITHM STEPS

According to proposed PCFG algorithm, we split the problem of parse tree generation into following phases [12].

- Tokenization: we transformed the data into suitable format. For processing simplicity we convert the Document by replacing the running text after certain threshold value by token.
- Label extraction: The algorithm first collects the labels, based on their alignment information into different sets representing different groups of labels.
- Calculate probabilities: For all labels (substring) calculate the probabilities of $w_1 \mid w_2 \ldots \mid w_n$ $[p_1, p_2 \ldots \ldots p_n]$ using equation $P(R_i)$.
- Parse tree: the $P(Ri)$ for every w_i generate the parse tree T of w_i as one of the node if it belongs to label set then as interior node else leaf node.
- Generating reduced parse tree: The reduced parse tree T' is obtained by deleting all the leaves of the tree. Which represents the reduced parse tree, of label sequence $l_1, l_2, \ldots . l_m$ corresponds the leaves.

VII. EXPERIMENTAL RESULT

For the experiment research papers of varying labels were chosen. In the first phase for document the labels were generated using pair wise alignment based approach and then complete parse trees were generated. For simplicity the maximum depth of parse tree was restricted to three. Consider the following example,

A. Sample Document

INTRODUCTION

Looking back into the past, back in 80's, mobile phones were a rare utility with less than 5 million subscribers worldwide. The phones available at that time were bulky having low battery life.

The Mobile Age

Mobile devices have revolutionized the life of every individual in some way or other. The current areas of usage as well as probable future utility of mobile technology can be illustrated as follows

Need for Mobile Technology

The last two decades have experienced a remarkable progress in the area of mobile technology both technologies wise as well as service wise.

Figure 1. Sample Document.

VIII. 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 [22].

- 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.

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

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 %
Document one	1000	81.3	79.6	80.4
Document two	10000	72.8	74.7	73.7
Average	---	**77**	**77.1**	**77**

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.

IX. CONCLUSION

We have demonstrated that a proposed algorithm with alignment similarities successfully used to extract information from the document stored as text document. There are several advantages of proposed algorithm. It can model hierarchical structure, in generalize form. The labeling of web documents can improve precision of Search Engine for information extraction. The proposed algorithm also allows for rich collection features which are utilized to measure the properties of sequence of tokens.

REFERENCES

[1] Montague, R. "The proper treatment of quantification in ordinary English". *In Thomason, R. H., editor, Formal Philosophy - Selected Papers of Richard Montague*, Yale University Press, New Haven:CT, USA and London, UK, chapter 8, pp 247–270, 1974.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Grandy, Richard E. "Understanding and the principle of compositionality." *Philosophical Perspectives* 4, pp 557-572, 1990.

[3] P. Dupont, L.Miclet and E. Vidal, "What is the search space of the regular inference?" *In Proc. ICGI (Lecture Notes in Artificial Intelligence), Heidelberg, Berlin: Springer,* vol. 862 pp 25–37, 1994.

[4] D. Angluin, "Queries and concept learning," *Mach. Learn.,* 2(4), pp 319–342, 1988.

[5] Y. Sakakibara, "Learning context-free grammars from structural data in polynomial time," *Theor. Comput. Sci.,* vol. 76, pp 223–242, 1990..

[6] D. Angluin, "Learning k-bounded context-free grammars," *Yale Univ.,New Haven, CT, Yale Tech. Rep.* RR-557, 1987.

[7] Y. Sakakibara, "Learning context-free grammars using tabular representations," *Pattern Recognit.,* vol. 38, no. 9, pp 1372–1383, 2005..

[8] Y. Sakakibara and M. Golea, "Simple recurrent networks as generalized hidden markov models with distributed representations," *In Proc. IEEE Int. Conf. Neural Network. New York: IEEE Comput. Soc.,* pp 979–984, 1995.

[9] K. Kersting, L. D. Raedt and T. Raiko, "Logical hidden Markov models," J. *Artif. Intell. Res.,* vol. 25, pp 425–456, 2006.

[10] P. Adriaans and M. Vervoort, "The EMILE 4.1 grammar induction toolbox," *In Proc. ICGI (Lecture Notes in Computer Science),* vol. 2484, pp 293–295, 2002.

[11] M. V. Zaanen, "Implementing alignment-based learning," In *Proc. ICGI (Lecture Notes in Computer Science),* vol. 2484, pp 312–314, 2002.

[12] Ramesh Thakur, Suresh Jain, Narendra S. Chaudhari, Rahul Singhai "Information extraction from semi-structured and un-structured documents using probabilistic context-free grammar inference," In *Proceedings* IEEE *International Conference on Information Retrieval & Knowledge Management (CAMP), 2012* Kuala Lumpur, pp 273-276 2012.

[13] Vinajak R. Borkar, K. Deshmukh, and Sunita Sarawagi, "Automatically extracting structure from free text addresses," *In Bulletin of the IEEE Computer Society Technical committee on Data Engineering, IEEE,* 2000.

[14] David M. P. Powers. "Machine learning of natural language". *Association for Computational Linguistics/European Chapter of the Association for Computational Linguistics Tutorial Notes, Madrid, Spain.* 1997.

[15] Andrew Kehler and Andreas Stolcke. "Unsupervised Learning in Natural Language Processing". *Association for Computational Linguistics. Proceedings of the workshop. In Preface A. Kehler and A. Stolcke, editors,* 1999.

[16] Chomsky N "Syntactic Structures", *The Hague Mouton,* 1975.

[17] P. Viola and M. Narasimhan, "Learning to extract information from semistructured text using a discriminative context-free grammar." *In Proceedings of the ACM SIGIR,* 2005.

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

[19] N. S. Chaudhari and X. Wang, "Language Structure Using Fuzzy Similarity," *IEEE transactions on fuzzy system,* vol. 17, no. 5, pp 1011-1024, 2009.

[20] M. V. Zaanen, "Theoretical and practical experiences with alignmentbased learning," *presented at the Australas. Lang. Technol. Workshop, Melbourne, Australia,* 2003.

[21] M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: Annotating predicate argument structure,"*In ARPA Human Language Technology Workshop, Plainsboro, New Jersey. Morgan Kaufmann, San Francisco,* pp 114–119, 1994.

[22] 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.