Expert System and Natural Language Inference

Himnandini Dubey[1]
[1]Department of Computer Science Engineering
[1]Al-Falah School of Engineering and Technology Dhauj, Faridabad, Hariyana

Abstract— Expert system has to gather data from the user in order to solve Problem.The high complexity of natural language and the huge amount of human and resources necessary for producing the research in the area of Natural Language Processing to investigate various solutions for automating grammar generation and updating processes. Many algorithms for Context-Free Grammar inference have been developed in the literature. This paper provides a survey of the methods for inferring context-free grammar. After introducing some definitions and presentation and the type of information formed. Moreover, the state of the art of the strategies for evaluation and comparison of different grammar inference methods is presented. The goal of the paper is to provide a reader with introduction to major concepts and current approaches in Natural Language Learning research.

Key words: Expert system, natural language inference, grammatical inference

I. INTRODUCTION

System that uses natural language as a means of communication must do in a natural manner. High-precision Natural Language Understanding is needed. Expert systems have become a popular method for representing large bodies of knowledge for a given field of expertise and solving problems by use of this knowledge [1].

Grammatical inference (also known as grammar learning) deals with idealized learning procedures for acquiring grammars on the basis of exposure to evidence about languages. Major areas of research in Natural Language Processing (NLP) many grammar inference algorithms have been developed in this. These algorithms have been differently classified by several authors, according to the various features of the inference process [2]. In this paper, a structured overview of the existing grammar inference methods for natural language is given. This survey provides a comprehensive summary of the present state of the literature, considering a wide scope of grammar inference methods. A classification of these methods is provided, based on the presentation set (e.g., text And informant) and the type of information (e.g., supervised,Unsupervised and semi-supervised) they use for learning the grammar [3].

II. EXPERT SYSTEM.

An expert system uses knowledge specific to a problem domain to provide “expert quality” to perform in a specific application area. An expert system is an interactive computer-based decision tool that uses both facts and rules to solve difficult decision problems based on knowledge acquired from an expert. An expert system often consists of three parts, namely: a knowledge base, an inference engine, and a user interface, A dialogue is conducted by the user interface between the user and the system. The user provides information about the problem to be solved and the system then attempts to provide insights derived (or inferred) from the knowledge base. These insights are provided by the inference engine after examining the knowledge base.

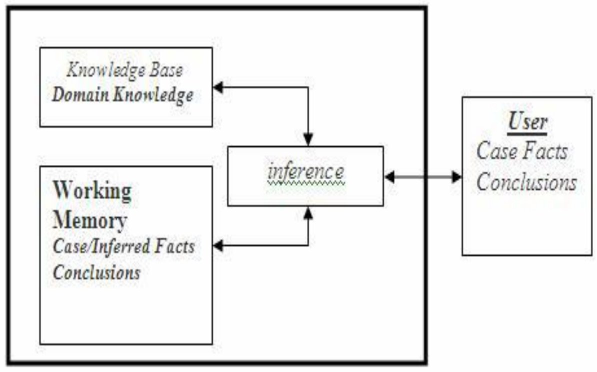

Figure 1. Expert System Components

III. THE INFERENCE ENGINE

In order to execute a rule-based expert system using the method of forward chaining we merely need to fire (or execute) actions whenever they appear on the action list of a rule whose conditions are true. This involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied.

Grammatical inference methods for natural language the main research studies in grammatical inference have been made in various application domains, such as speech recognition), computational linguistic, Computational biology and machine learning.

The majority of natural language learning methods, proposed in the literature, is based on a text-based and unsupervised approach. The reason for that is explain in the three points.

(1) Unsupervised learning enables to learn larger and more complex models than supervised learning. This is because supervised learning aims at defining connections between one set of input sentences and another set of output sentences (provided by the structured corpus). Therefore, the complexity of the learning task increases notably when learning models with deep hierarchies.
(2) Supervised methods typically generate better results, due to the fact that they know the structure of the language for tuning their output, unsupervised methods are less time-consuming and costly because they do not require the onerous creation of the initial tree-bank of the language.
(3) Natural Language is problematic to specify all sentences that have not to be included into the grammar because learners typically get evidence about what is grammatical (correct sentences or positive examples), but no details about what is not

grammatical (incorrect sentences or negative examples).

IV. DESCRIPTION OF METHODS

The grammatical inference methods for Natural Language.

A. ADIOS

The ADIOS (Automatic Distillation of Structure) algorithm was proposed as a statistical method of grammar induction that yields symbolic results in the form of a context-free grammar. It induces grammars from a corpus of strings (such as text, transcribed speech, nucleotide base pairs, etc.), using only positive examples in an unsupervised fashion. Therefore, ADIOS can be classified as a text-based and unsupervised grammatical inference method.[9].The algorithm works in three phases: initialization, pattern distillation and generalization. Initialization involves loading the corpus onto a directed pseudo graph (i.e. a non-simple graph in which both loops and multiple edges are permitted) whose vertices are all lexicon entries, augmented by two special symbols, `begin` and `end`. A sentence in the graph is represented by a path over the graph, starting at `begin` and ending at `end`, and is indexed by the order of its appearance in the corpus. In the top of Fig. 2, the pseudo graph for the sentence "that the cat is eager to please disturb Beth" is initialized. Initialization is followed by pattern distillation that consists in the extraction of significant patterns (i.e. sequences of nodes) from the pseudo graph by finding sub-paths of high probability, considering the number of outgoing and incoming edges of a sub-path. After the candidate patterns have been generated, the generalization phase looks for finding the most significant pattern, then generalizes over the graph by creating equivalence classes from all of the variable node elements in the pattern. For instance, in the equivalence class consisting of the nouns {*bird, cat, cow, dog, horse, rabbit*} is created. At the end of each iteration, the most significant pattern is added to the lexicon as a new unit, the sub-paths it subsumes are merged into a new vertex, and the graph is rewired accordingly.

B. Self training

The self-training method was firstly proposed by Charniakand inspired several subsequent works (McClosky). It learns syntactic structures from both a small set of declarative labeled examples, which are used to train an initial model, and a larger set of declarative unlabeled examples, which are labeled by the trained model and used for re-train a new model. Therefore, the self-training method belongs to the class of text-based and semi-supervised grammatical inference methods. The self-training method proposed by Charniak.Applies a probabilistic model that uses a CFG for specifying how the unlabeled sentences can be parsed and which is the prob-ability of the possible parses. Before applying this model for parsing unlabeled sentences, the parser is trained by using the small set of labeled sentences. Afterwards, the parsing of the unlabeled sentences is carried out by the trained parser applying the probability model. It extended the method by introducing a further step, in which a discriminative reran Ker reorders the possible parses of each unlabeledsentence according to several features of the parses, defined in another work of the same authors (Charniak and Johnson 2005). Roughly, each feature f_j is a function that maps a parse y to a real number. The feature's value $f_j(y)$ is the number of times that the feature occurs for the parse y. For example, the feature $f_{eat\ pizza}(y)$ counts the number of times that a phrase in y, headed by eat, has a complement phrase, headed by pizza.

C. Emile

EMILE was firstly proposed by Adriaans in 1992 and successively updated through the years until the latest version. It is based on a teacher/pupil metaphor, where the teacher generates grammatically correct sentences (positive examples) and the pupil can ask the oracle which one is valid. Therefore, EMILE belongs to the class of *text-based* and *supervised* grammatical inference methods.

More in detail, the EMILE algorithm consists of five main steps: first order explosion, verification, clustering, rule induction and rule rewriting. Given an input set of positive example sentences, each sentence is examined to discover how it can be broken up into sub expressions. For instance, considering the example sentences "John loves Mary" and "Mary walks", a possible set of sub expressions is the following: {"John", "John loves", "John loves Mary", "loves Mary", "Mary", "and Mary walks ","" walks"}. The resulting set of sub expressions is passed to an oracle to verify substitutions of all expressions in each context. The oracle returns whether or not each subespression is a valid sentence. Afterwards, the next stage consists in clustering context rules (passed by the oracle) into types. Expressions that can be substituted into the same contexts belong to the same type. For instance, the context rules S/ loves Mary → J john and S/ loves Mary → Mary imply that John and Mary can be substituted in the same context and therefore they belong to the same type A. In the rule induction phase, the basic and complex rules, associated to specific types during the clustering phase, are generalized toward more general types and consequently new grammar rules are introduced. For, instance, the grammar rules S/ loves Mary → A and A → John | Mary are introduced. Finally, the rules are rewritten with the outcome of producing context-free grammar rules.

V. EVALUATION OF GRAMMAR INFERENCE METHOD

The evaluation of grammar inference algorithms is not a trivial task, and many different approaches have been proposed in the literature.

VI. EVALUATION STRATEGY

An approach for evaluating grammar inference algorithms is the "*Compare againstTreebank*". This evaluation method consists in applying the grammar inference algorithm toa set of plain natural language sentences that are extracted from an annotated Treebank, which is selected as a "gold standard". The structured sentences generated by the algorithm are then compared against the original structured sentences from the Treebank. A schema describing how a learning system is evaluated against a Treebank is shown . There are several metrics that can be used to compare the learned tree against the original tree structure. Most often, the recall, which gives a measure of the completeness of the learned grammar, and the precision, which shows how correct the learned structure is, are used. These two metrics, along with the crossing brackets, which is a metric counting the number of response constituents that violate the boundaries of a constituent in the key, belong to the PARSEVAL scoring metrics, proposed by Black et al.for comparing a candidate parse (the output of the algorithm) with its reference parse from the annotated corpus. Another metric is the f_1 score, which can be interpreted as a weighted average of the precision and recall metrics. The "*Compare Against Treebank*" method does not need an expert to indicate if some construction is correct or incorrect, allowing for a relatively objective comparison of different algorithms. The main problem with this approach is that structured corpora are needed.

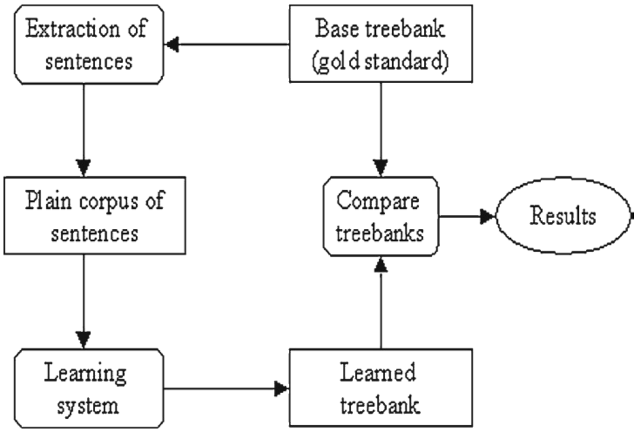

Fig. 2: Functioning of the "Compare Against Treebank" strategy.

VII. CONCLUSION

In this survey, we discussed the problem of natural language learning and gave an overview of the existing grammar inference methods for natural language.

We took into account the kind of presentation (if text or in format) and the type of information (if supervised, unsupervised, or semi-supervised) to broadly classify the grammar inference methods for natural language into *informant-based* and *text-based* methods, and *supervised*, *unsupervised*, and *semi-supervised* methods, respectively.

The introduced methods have been then analyzed considering how they have evolved in time and taking into account their underlying computational techniques.

In particular, an analysis of scientific production, based on bibliographic data, has been carried out for obtaining indicators about temporal evolution, variations and trends in the field of grammar inference methods.

The current state of the art includes lots of underlying computational techniques applied to perform the language learning: statistical methods, evolutionary computing techniques, minimum description length, heuristic methods, greedy search methods, clustering techniques. An overview of the techniques that are applied in each of the investigated grammar inference methods has been provided in the paper. the majority of NL learning methods, proposed in the literature, is based on an unsuper-vised approach. This is due to the fact that unsupervised learning enables to learn larger and more complex models than supervised learning and they are also less time-consuming and costly because they do not require the onerous creation of the initial tree-bank of the language.

REFERENCES

[1] Computational linguistics (Adriaans 1992)
[2] Baker JK (1979) Trainable grammars for speech recognition.
[3] Black E, Abney S, Flickinger D, Gdaniec C, Grishman R, Harrison P, Hindle D, Ingria R, Jelinek F, Klavans J, Liberman M, Marcu M, Roukos S, Santorini B, Strzalkowski T (1991) A procedure for quantita-tively comparing the syntactic coverage of English grammars. In: Proceedings of the DARPA speech and natural language workshop, pp 306–311
[4] Blum A, Mitchell T (1998) Combining labeled and unlabeled data with cotraining. In: Proceedings of the workshop on computational learning theory.
[5] Bonnema R, Bod R, Scha R (1997) A DOP model for semantic interpretation. In: ACL 1997, pp 159–167 Briscoe T (2000) Grammatical acquisition: inductive bias and coevolution of language and the languageacquisition device. Language Charniak E, Johnson M (2005) Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In: Proceedings of the 43rd annual meeting of the ACL, Ann Arbor, pp 173–180
[6] Charniak E (1997) Statistical parsing with a context-free grammar and word statistics. In: Proceedings of the fourteenth national conference on artificial intelligence, Menlo Park. AAAI Press/MIT Press
[7] Chomsky N (1957) Syntactic Structures. The Hague Mouton.
[8] computational biology (Sakakibara et al. 1994
[9] Salvador and Benedı 2002), and ma-chine learning (Sakakibara 1997;(7) de la Higuera and Oncina 2003)(8) (Solan et al. 2005) .
[10] Hopcroft JE, Ullman JE (1979) Introduction to automata theory, languages, and computation. Addison-Wesley, New York
[11] Horning JJ (1969) A study of grammatical inference. PhD thesis, Stanford University, Stanford:CA, USA Kasami T (1965) An efficient recognition and syntax analysis algorithm for context-free languages. Science report, Air Force Cambridge Research Laboratory, Bedford (Baker 1979), Angluin D (1982) Inference of reversible.

