PRES LUNAM
Ecole Doctorale STIM
Sciences et Technologies de l'Information et Mathématiques

Spécialité : Informatique
Laboratoire : LINA
Equipe : TALN

Grammatical Inference of Probabilistic Context-Free Grammars

James Scicluna
Mél : james.scicluna@univ-nantes.fr

Abstract:
Syntax analysis, the process of assigning structural descriptions to natural language sentences, is one of the central tasks in natural language processing. It is a necessary step for semantic analysis, which can be used in various domains including machine translation, speech recognition and question answering. The goal of our research is to get state-of-the-art results for syntax analysis by taking a grammatical inference approach. With this approach, we try to describe an algorithm that learns a grammar from information about the natural language, and use this grammar as the mechanism for assigning structural descriptions to sentences. We chose to learn probabilistic context-free grammars, as they are adequate models for natural language syntax, in an unsupervised setting. Our aim is to solve the learning problem by first giving a theoretical identification result and then building upon this a robust practical learning algorithm that gives good results in practice.

Keywords: *Grammatical Inference, Probabilistic Context-Free Grammars, Natural Language Processing, Syntactic Analysis, Machine Learning, Identification*

1 Problem Definition

Natural language processing (NLP) is a multidisciplinary field combining computer science, linguistics, artificial intelligence and mathematics. The goal of this field is to build computer programs which are able to perform tasks related to human languages. Examples of such tasks include:

- Machine Translation: Translating spoken words or text from one language to another. Google Translate is an example of a free machine translation tool.
- Speech Recognition: Translating spoken words into text and/or machine interpretable commands. Nowadays, smart phones are equipped with speech recognition tools that recognize the user's spoken commands (e.g. "Call Tom", "Open Message Inbox")
- Question Answering: Systems which understand and answer questions using information from databases or the internet. Google makes use of question answering tools whenever the user inputs a question like "What is the capital city of Australia?".
- Automatic Summarization: Automatic production of summaries from long texts that retain the most salient points. This can be very useful for students before an exam!
- Sentiment Analysis: Automatic extraction of the attitudes and emotions (e.g. positive, negative, indifferent) from large amount of texts, normally applied on online reviews and social media comments. This is very useful in public relations and marketing, where companies are interested in knowing the public's perceptions of their organization.

For all of these tasks to be possible, the computer must be able to have a **semantic interpretation** of natural languages. This means that the computer must not treat natural language sentences simply as sequences of symbols, but it must give **meaning** to sentences. However, in order for a computer to give semantic interpretation, it must first be able to find the **structure** in language sentences. The structure of a sentence shows how different parts of a sentence relate to each other, like for example subject-verb-object. The process of finding such structure is called **syntactic analysis**. Figure 1 shows different **structural descriptions** that can be given to a sentence through this process.

The goal of our research is to come up with novel ways of performing syntactic analysis. We specifically focus on finding **phrase structure trees** (structure (b) in Figure 1).

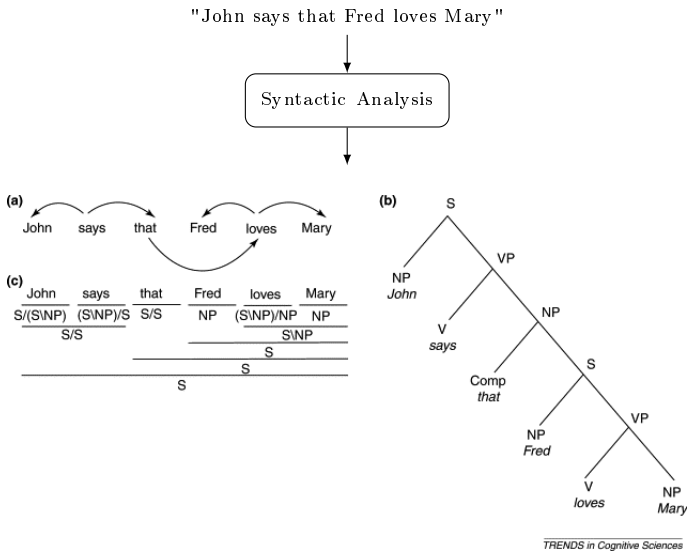


Figure 1: Different structural descriptions a syntactic analysis process can give. Structure (a) is a dependency structure, structure (b) is a phrase structure tree and (c) is a categorial structure (reproduced from [1])

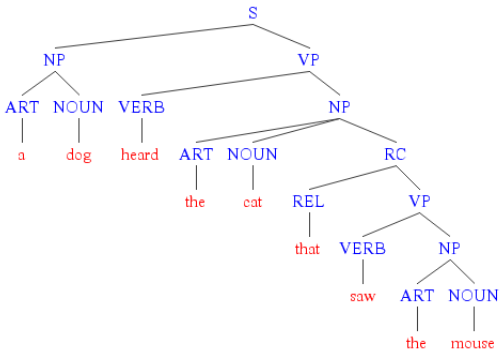
2 Approach

We take a **grammatical inference** [2] approach to solve the syntactic analysis problem. This means that we build a computer program that is able to **learn** a **grammar** from **information** about the natural language and use it as the mechanism for assigning phase structure trees to sentences. The keywords here are *learning*, *grammar* and *information*. We'll explain each in the following subsections:

2.1 The Grammar

A grammar is just a set of rules that describe how to form a sentence. The path taken when following the rules to build the sentence will determine the structure of the sentence. The following is a grammar (to the left) and the structure it generates for the sentence *"a dog heard the cat that saw the mouse"* (to the right):

S → NP VP
VP → V NP
NP → ART NOUN
NP → ART NOUN RC
RC → REL VP
VERB → saw | heard
NOUN → cat | dog | mouse
ART → a | the
REL → that



This type of grammar is called a **context-free grammar** [3]. In our work, we use a probabilistic version of this type of grammar. This is because it is more feasible in practice to work in a probabilistic setting for various technical reasons.

2.2 The Information

There are 3 main types of information about the natural language that can be used for learning a grammar:

1. Big treebanks (i.e. a big collection of sentences annotated with phrase structure trees)

2. Small treebanks + unlabelled sentences (i.e. normal sentences without structure)

3. Unlabelled sentences only

In our work, we opt for the third option (which is known as the **unsupervised setting**). This is because in reality treebanks are scarce and too costly to build, especially for under-resourced languages. Moreover, for certain applications of our work, it is inadequate to use treebanks. Also, the trend in the research community is to work with unlabelled sentences only.

2.3 Learning

There are two approaches we can take to solve the learning problem: either get a theoretical learning algorithm which provable works or build a heuristic-based learning algorithm. The problem with the theoretical learning algorithm is that it is impossible to describe one under no restrictions on the information given and the grammars to be learned [4]. On the other hand, blind heuristic-based algorithms do not give the best results in practice [2].

So, the approach we take is to start from a theoretical learning algorithm in a restricted setting (which is not applicable in practice) and then build a practical learning algorithm as a robust version of the theoretical one. The steps we take to this are:

1. Describe a restricted class of grammars and conditions on the information given

2. Describe a theoretical learning algorithm for this restricted class

3. Prove that this algorithm is correct

4. Enhance the theoretical learning algorithm to make it robust enough to be applied in practice

References

[1] Timothy J. O'Donnell, Marc D. Hauser, and W. Tecumseh Fitch. Using mathematical models of language experimentally. *Trends in Cognitive Sciences*, 9(6):284 – 289, 2005.

[2] Colin de la Higuera. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, New York, NY, USA, 2010.

[3] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[4] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.