Flexible Module for Shallow Parsing, Using Preferences

Fernando M. Batista*, Nuno J. Mamede†

*L²F /INESC-ID /ISCTE †L²F /INESC-ID /IST
Spoken Language Systems Lab
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
{Fernando.Batista, Nuno.Mamede}@inesc-id.pt

Abstract

This paper presents a shallow parsing module – SuSAna – that performs efficient analysis over unrestricted text. The module recognizes the boundaries, internal structure, and syntactic category of the syntactic constituents. In addition to the definition of syntactic structures, its grammar supports a hierarchy of symbols and a set of restrictions known as preferences. During the analysis, a directed graph is used for representing all the operations, preventing redundant computation. The algorithm has $O(n^2)$ complexity, where n is the number of lexical units in the segment. SuSAna can be used as a standalone application, fully integrated in a larger system for natural language processing, or in a client/server platform.

1. Introduction

The syntactic analysis of a corpus returns information otherwise hidden, allowing the development of more powerful and complex applications. The syntactic processing of corpora may be applied to areas such as information retrieval, information extraction, speech synthesis and recognition (Marcus Fach, 1999) and automatic translation. Syntactic analysis is also frequently the starting point for semantic processing systems.

The shallow parsing module, SuSAna (Surface Syntactic Analyzer), performs efficient analysis over unrestricted text. The development of the module is based on the work of Caroline Hagège (2000), and recognizes, not only the boundaries, but also the internal structure and syntactic category of syntactic constituents. Its grammar supports a hierarchy of symbols and a set of restrictions known as *preferences* (Tomek Strzalkowski, 1994), in addition to the definition of the syntactic structures. During the analysis, a directed graph is used for representing all the operations, preventing redundant computation. The algorithm has $O(n^2)$ complexity, where n is the number of lexical units in the segment. SuSAna can be used as a standalone application, fully integrated in a larger system for natural language processing, or in a client/server platform.

2. The knowledge base

The structures SuSAna identifies, known as *models*, are defined from a set of properties. In the scope of the analysis, morphossyntactic categories are also viewed as models, thus the concepts of *terminal model* and *non-terminal model* are used to distinguish the categories from the models.

The grammar structure defined for SuSAna has been adapted and improved from the grammar used by the shallow parsing prototype AF (Caroline Hagège, 2000). This grammar uses three different structures for representing all the lexical information: block structures define the behavior of models inside other models; *preferences* are used for choosing between different interpretations, according to confidence levels; and a symbol hierarchy, that alllows to define classes and subclasses of models, leading to a clear and reduced number of rules.

Besides preferences, SuSAna makes use of psycholinguistic principles (Daniel Jurafsky and Martin, 2000; Allen, 1995), for choosing between different interpretations that the parser might be able to find. Currently, the module uses the longest model principle, which states that all other things being equal, new constituents tend to be interpreted as being part of the constituent under construction rather than part of some constituent higher in the parse tree. In the future other psycholinguistic principles, such as minimal attachment and right association, may be applied.

3. Algorithm and internal organization

3.1. Architecture

The overall analysis process is performed in two stages. The first stage consists of generating the information concerning the input data and storing it into a repository. The repository will then provide, in a second stage, all the information required for producing the desired output. As shown in Figure 1, the analysis and extraction tasks are performed independently and can be independently parametrized. Besides providing all required data to the extraction module, the repository saves information about previous calculations, thus preventing redundant computation.

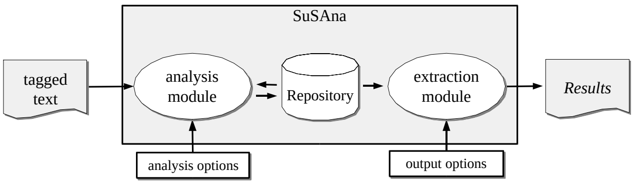

Figure 1 - SuSAna's internal architecture.

3.2. The algorithm

In order to cover unusual linguistic constructions, the algorithm finds all possible sequences for the analysis during the first phase, then selects the most promising ones, either according to preferences or by applying psycholinguistic principles.

The analysis of a given sentence is represented using an in-memory DAG (*Directed Acyclic Graph*). Each vertice of the graph is associated with a lexical unit of the sentence and contains information about the occurrence of a model inside other model, in that position of the sentence. The DAG makes use of two types of edges, one for specifying child vertices and the other for specifying sibling vertices. Each edge has an associated cost, given by the preferences specified in the grammar. The analysis consists of, being at a given vertice, finding all possible child vertices and, when done, finding all sibling vertices. Whenever possible, the algorithm reuses previously calculated analysis fragments, achieving results faster.

Selection of the most promising paths consists of ranking paths from the starting point of the graph, based on the cost associated with each edge and on the longest models principle. The full paper will describe the employed strategy in detail.

The algorithm is robust, in the sense that it can skip unexpected, or out of context, lexical units and reduce as much as possible the number of hypotheses for each analysis, thus providing output suitable for further processing. Special grammar rules may be introduced, in order to increase the robustness.

4. Parametrization

The previously presented architecture allows a flexible way of setting analysis and extraction options. In what concerns analysis options, one of the most important is the possibility of defining the starting model, overriding the default one, during execution. Another important option is the possibility of skipping untreatable lexical units at the beginning and at the end of the analysis, making it possible to find the best solution without considering those words. This option can be used to find large linguistic structures in the segment when boundaries are not feasible. By default, each segment corresponds to a linguistic structure. However, it is possible to search for multiple linguistic structures in a segment, allowing, for example, the identification of sentences in a paragraph. This option can be used simultaneously with the option for skipping models, in order to extract all the linguistic structures of some type in a given segment.

Another interesting option for SuSAna is the ability to process incomplete structures. This is useful when there are no solutions and the user wants to know the largest analysis found. This can also be applied to guess, for an incomplete sentence, the categories that may follow the last lexical unit, so that the sentence remains correct according to the grammar.

5. Evaluation

In what concerns linguistic correctness, at the moment, only small tests have been performed, but they show promising results. The grammar currently in use was written by Caroline Hagège (2000) for extracting noun phrases. Linguistic phenomena, such as verb phrases, are superficially treated, preventing a full linguistic evaluation of the system. Nevertheless, comparisons between SuSAna and AF show better accuracy for SuSAna.

Tests were conducted over a corpus of about 4.6 million words, consisting of two months of the newspaper Público (Batista, 2003). In what concerns performance results in terms of processing time, SuSAna performed all the analyses at an average of about 300 words/second[1]. In what concerns coverage, 61.6% - 97.7% of the lexical units were covered by the analysis process, depending on the performed test. The value 61.6% corresponds to identifying the structure of previously segmented text, considering that each word was correctly placed in the segment. Using SuSAna to segment the corpus, 97.7% of the lexical units were covered.

6. References

Allen, J. (1995). *Natural Language Undestanding*. Benjamin/Cummings, Redwood City, CA, 2nd edition.

Batista (2003). *Análise Sintáctica de Superfície*. MSc Thesis. Universidade Técnica de Lisboa – Instituto Superior Técnico. Lisbon, Portugal. July 2003.

Fach, M. (1999). A comparision between syntactic and prosodic phrasing. In *proceedings of Eurospeech 1999*, volume 1, pages 527–530, Budapest.

Hagège, C. (2000). *Analyse Syntaxique automatique du portugais.* PhD thesis, Laboratoire de Recherche sur le Language, Université Blaise Pascal, Clermond-Ferrand, GRIL.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing.* Prentice Hall.

Strzalkowski, T., editor (1994). *Reversible Grammar In Natural Language Processing*. Kluwer Academic Publishers, Boston, London.

[1] Intel Pentium III processor at 800 Mhz. Linux operating system