Automatic Extraction of Grammars From Annotated Text

Salim Roukos, Principal Investigator

roukos@watson.ibm.com
IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Project Goals

The primary objective of this project is to develop a robust, high-performance parser for English by automatically extracting a grammar from an annotated corpus of bracketed sentences, called the Treebank. The project is a collaboration between the IBM Continuous Speech Recognition Group and the University of Pennsylvania Department of Computer Sciences[1]. Our initial focus is the domain of *computer manuals* with a vocabulary of 3000 words. We use a Treebank that was developed jointly by IBM and the University of Lancaster, England.

In this past year, we have demonstrated that our automatically built parser produces parses without crossing brackets for 78% of a blind test set. This improves on the 69% that our manually built grammar-based parser [1] produces. The grammar had been crafted by a grammarian by examining the same training set as the automatically built parser over a period of more than 3 years.

Parsing Model

Traditionally, parsing relies on a grammar to determine a set of parse trees for a sentence and typically uses a scoring mechanism based on either rule preference or a probabilistic model to determine a preferred parse. In this conventional approach, a linguist must specify the basic constituents, the rules for combining basic constituents into larger ones, and the detailed conditions under which these rules may be used.

Instead of using a grammar, we rely on a probabilistic model, p(T|W), for the probability that a parse tree, T, is a parse for sentence W. We use data from the Treebank, with appropriate statistical modeling techniques, to capture implicitly the plethora of linguistic details necessary to correctly parse most sentences. In our model of parsing, we associate with any parse tree a set of bottom-up derivations; each derivation describing a particular order in which the parse tree is constructed. Our parsing model assigns a probability to a derivation, denoted by p(d|W). The probability of a parse tree is the sum of the probability of all derivations leading to the parse tree. The probability of a derivation is a product of probabilities, one for each step of the derivation. These steps are of three types:

- a tagging step: where we want the probability of tagging a word with a tag in the context of the derivation up to that point.
- a labeling step: where we want the probability of assigning a non terminal label to a node in the derivation.
- an extension step: where we want to determine the probability that a labeled node is extended, for example, to the left or right (i.e. to combine with the preceding or following constituents).

The probability of a step is determined by a decision tree appropriate to the type of the step. The three decision trees examine the derivation up to that point to determine the probability of any particular step.

The parsing models were trained on 28,000 sentences from the Computer Manuals domain, and tested on 1100 unseen sentences of length 1 - 25 words. On this test set, the parser produced the correct parse, i.e. a parse which matched the treebank parse exactly, for 38% of the sentences. Ignoring part-of-speech tagging errors, it produced the correct parse tree for 47% of the sentences.

Plans for the Coming Year

We plan to continue working with our new parser by completing the following tasks:

- implement a set of detailed questions to capture information about conjunction, prepositional attachment, etc.
- improve the speed of the search strategy of the parser.

References

1. Black, E., Jelinek, F., Lafferty, J., Magerman, D. M., Mercer, R., and Roukos, S., 1993. Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In Proceedings of the Association for Computational Linguistics, 1993. Columbus, Ohio.

[1] Co-Principal Investigators: Mark Liberman and Mitchell Marcus