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, during the past three years.

RECENT RESULTS

We have an initial implementation of our parsing model where we used a simple set of features to guide us in our development of the approach. We used for training a Treebank of about 28,000 sentences. The parser's accuracy on a sample of 25 new sentences of length 7 to 17 words as judged, when compared to the Treebank, by three members of the group, is 52%. This is encouraging in light of the fact that we are in the process of increasing the features that the parser can look at. We give below a brief sketch of our approach.

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 (or some higher level processing is expected to do further disambiguation). 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. Once we have built our model, we parse a sentence by simply determining the most probable parse, T^*, for the given sentence W from the set of all trees that span the given sentence.

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.

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.
- build automatically a new set of classes for the words in our vocabulary.
- tune the search strategy for the parser.

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