Name:

Statistical Parsing & Disambiguation

Tutorial 6

The Parser Files

- The files and programs you will need are in:
  `/proj/courses/comppsych/`**`Tutorial6`**`/`
  Copy the entire directory to your account.

For this tutorial we will experiment with a top-down, incremental probabilistic parser developed by Brian Roark. While not explicitly developed as a cognitive model, it allows us to examine the kinds of behavior the parser exhibits for some of the lexical, structural, and combined ambiguities we've discussed in the lecture. Generally speaking, it is similar to the Jurafsky model, though it doesn't use detailed subcategorization frame probabilities, nor is it lexicalized.

1. Using the Roark Parser.

The Roark parser has already been installed and trained on the Coli computers. By using the following commands, it will parse all the sentences that occur in an input file, `test.txt`, and write them to an output file, `parse.output`

a) Among the text files from the `Tutorial8` directory, look at the content of the file `test.txt`

b) Create a symbolic link to the parser:
```
ln –s /proj/contrib/tdp.distrib/bin/tdparse tdp
```
c) Run the parser, and look at the output file:
```
./tdp -v -F test.output /proj/contrib/tdp.distrib/parse.model.slc.p05
test.txt
```
d) To determine the possible flags, and what they do, try:
```
./tdp -?
```
e) In order to use the tree view in (f), you will need to have an Xserver running, on your own machine.

f) To view the trees, you can run the `stanford-tregex-2012-07-09.jar` tool. Select your `test.out` file, and click on browse.

2. PP Attachment

To see how the parser ranks multiple possible global parses, use the `–k 2` switch, which outputs the k best parses. Try this for the PP attachment sentences in pp-attach.txt:

```
./tdp -k 2 -v -F pp-attach.out
/proj/contrib/tdp.distrib/parse.model.slc.p05 pp-attach.txt
```

Use the `stanford-tregex-2012-07-09.jar` tool. Select your `pp-attach.out` file, and click on browse, in order to look at the trees more comfortably.

What do you notice about the structural attachment preferences of the parser? Does it appear to exploit verb frame information?

3. Noun-Verb Category Ambiguity

The file nn-nv.txt has a number of the temporary N/V category ambiguities that we discussed in class. Run the parser as above, and see how it behaves. Note down the key points.

4. Reduced Relative Clauses

Now recall the ambiguous verbs in (reduced) relative clauses (file `relatives.txt`). How does the parser perform on these sentences?

5. Other structures

The file bonus.txt has a number of other example sentences that we have considered in class. Again, run the parser on these, and comment on where the parser goes wrong, or prefers the wrong parse over the correct one.