In class exercise: a garden-path ambiguity and probabilistic grammar

9.19(0) Fall 2023, Instructor: Roger Levy

18 October 2023

Syntactic ambiguities

One of the well-studied **local syntactic ambiguities** in English is exemplified by the following string prefixes:

(1) Jamie heard the...

(2) Jordan claimed the...

This ambiguity is sometimes called the "NP/S" ambiguity (Sturt et al., 1999), referring to whether the complement of the verb is an NP or an S category (in the Penn Treebank, it's an SBAR category that contains an S; SBAR is the category used in that corpus for complement clauses, subordinate clause, and relative clauses).

This ambiguity can be prevented when the word *that* comes between the verb and the beginning of the following NP, e.g.:

(3) The broadcaster announced that the...

Write a grammar fragment that captures the syntactic ambiguity and how it can be definitively resolved by following material. Use Tregex searches to estimate the probabilities of the relevant rules from an English treebank corpus. Would your grammar initially favor the NP interpretation or the S interpretation?

Next: In an influential psycholinguistics paper, Garnsey et al. (1997) showed that the syntactic "bias" of the verb affects continuation preferences. What are your intuitions about the respective biases of *heard* and of *claimed*? Use tree searches to find the empirical relative frequencies. Do they match your intuitions? How would we incorporate these verb-contingent differences in phrase probabilities into our grammar fragment?

Finally: In the ambiguous cases of Examples (1)–(2), the word after *the* can push around preferred interpretations even without categorically resolving the ambiguity. Can you think of examples like this? Very broadly, how might we consider handling them within a theory of human incremental parsing and structural disambiguation?

Some of the Tregex language

If you do a web search for "TregexPattern" the first hit you get should be the documentation for Tregex:

https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/trees/tregex/TregexPattern.html

A couple of crucial notes:

- You can and should use parentheses to make clear what relations you want!
- In a chain of relations A op B op C ..., all relations are relative to the first node in the chain. For example, `(S < VP < NP)` means "an S over a VP and also over an NP". Nodes can be grouped using parentheses '(' and ')' as in `S < (NP $++ VP)` to match an S over an NP, where the NP has a VP as a right sister. So, if instead what you want is an S above a VP above an NP, you must write `S < (VP < NP)`
- If you wrap a node label with '/' marks, it is interpreted as a regular expression, e.g., `/^hear/` matches any node that begins with the string `hear` (including heard, hearing, hearable, ...)
- A double underscore, `__`, matches any node.
- Relations (e.g., `< NP`) can be disjoined using `|` and negated using `!`. (They can also be made optional using `?` but this is not relevant except for more advanced usage.)
- Penn Treebank node labels often have functional tags introduced by `-` and `=`, like `NP-SBJ`. Often we want to ignore these in tree searches. Prepending a node label with `@` will make Tregex ignore these functional tags for matching purposes.

Here are a few relations that you may find especially useful for this exercise and for the pset:

```
Relation   Meaning
A << B     A dominates B
A < B      A immediately dominates B
A <i B     B is the ith child of A (i > 0)
A <: B     B is the only child of A
A >: B     A is the only child of B
A $+ B     A is the immediate left sister of B
A <<# B    B is a head of phrase A
A <# B     B is the immediate head of phrase A
```