More about PCFG parsing

MAS.S60

Catherine Havasi

Rob Speer

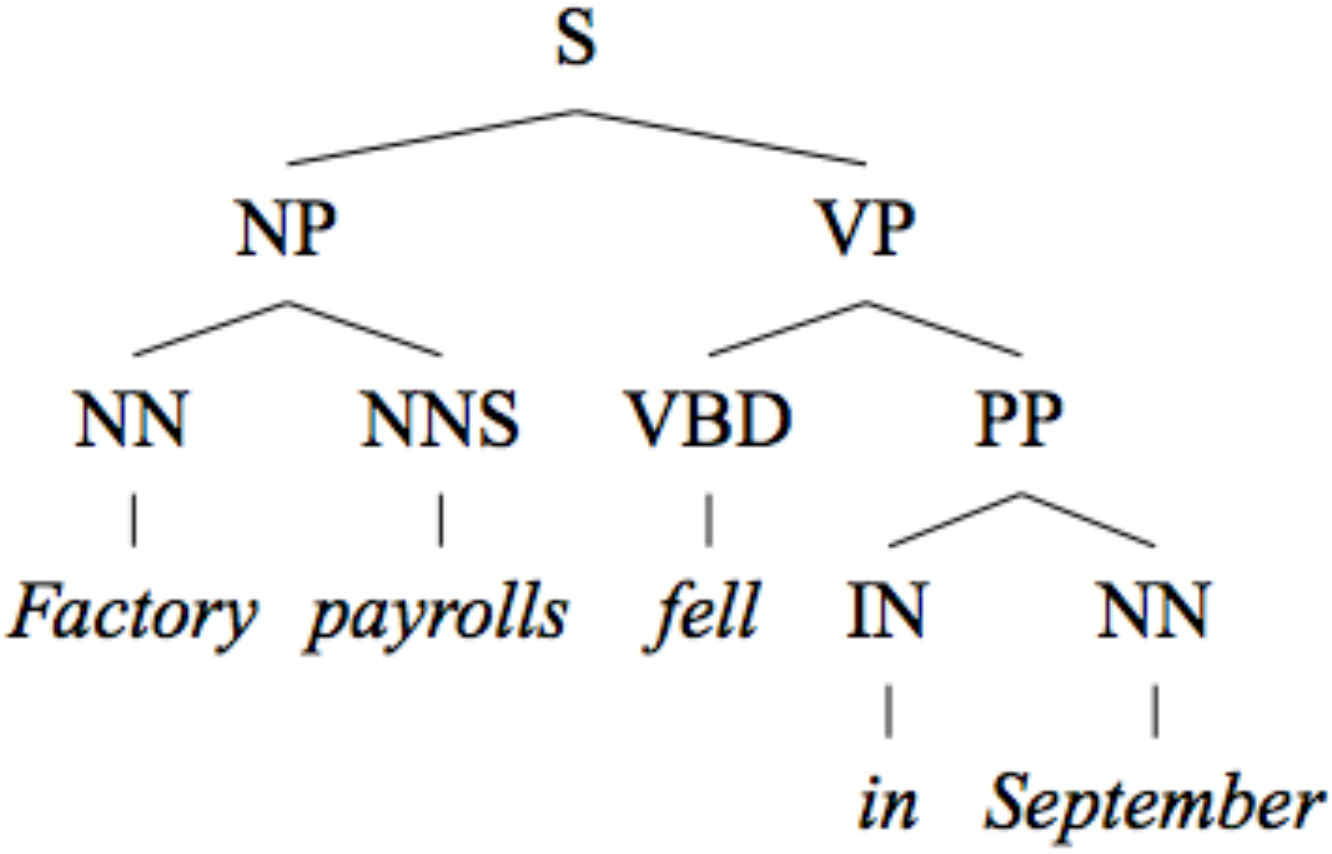
Summary from last time

- Context-free grammars match the principles of how language works, but are too simple to actually parse a sentence
- Probabilities help
  - and they let us use the greedy, bottom-up CYK algorithm
- Adding lexical information helps
  - VP(eat) attaches to PP(with)
  - VP(eat) doesn't attach to PP(of)

How the Stanford parser does it

(a brief overview)

Start with a PCFG parser



Modify it to include some context

- Take into account extra features such as each node's parent, neighbors, and number of words

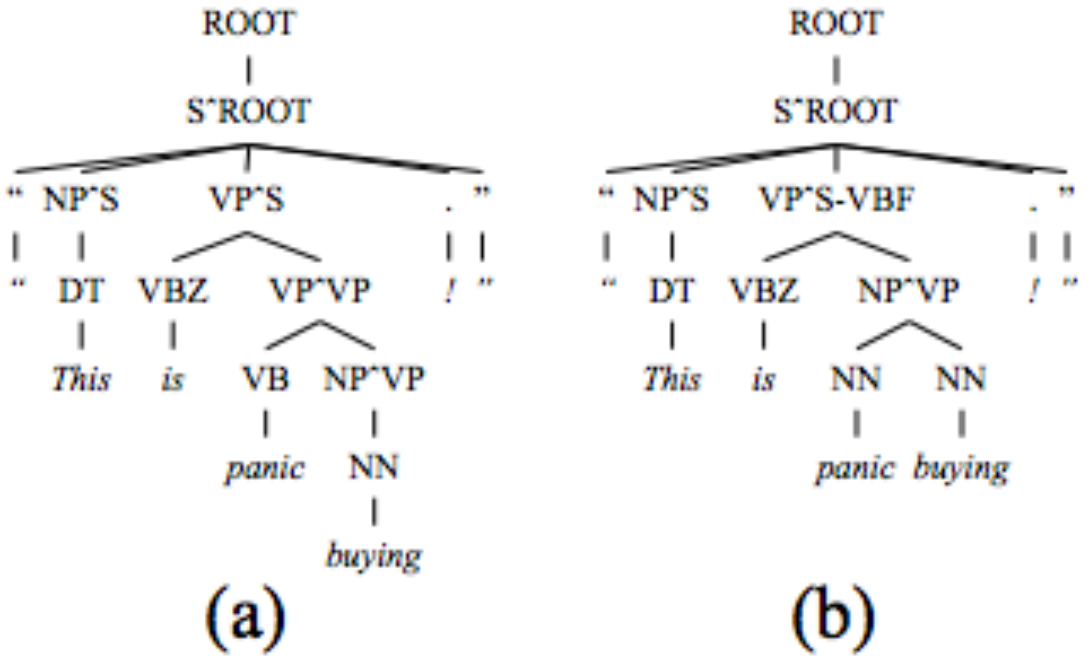
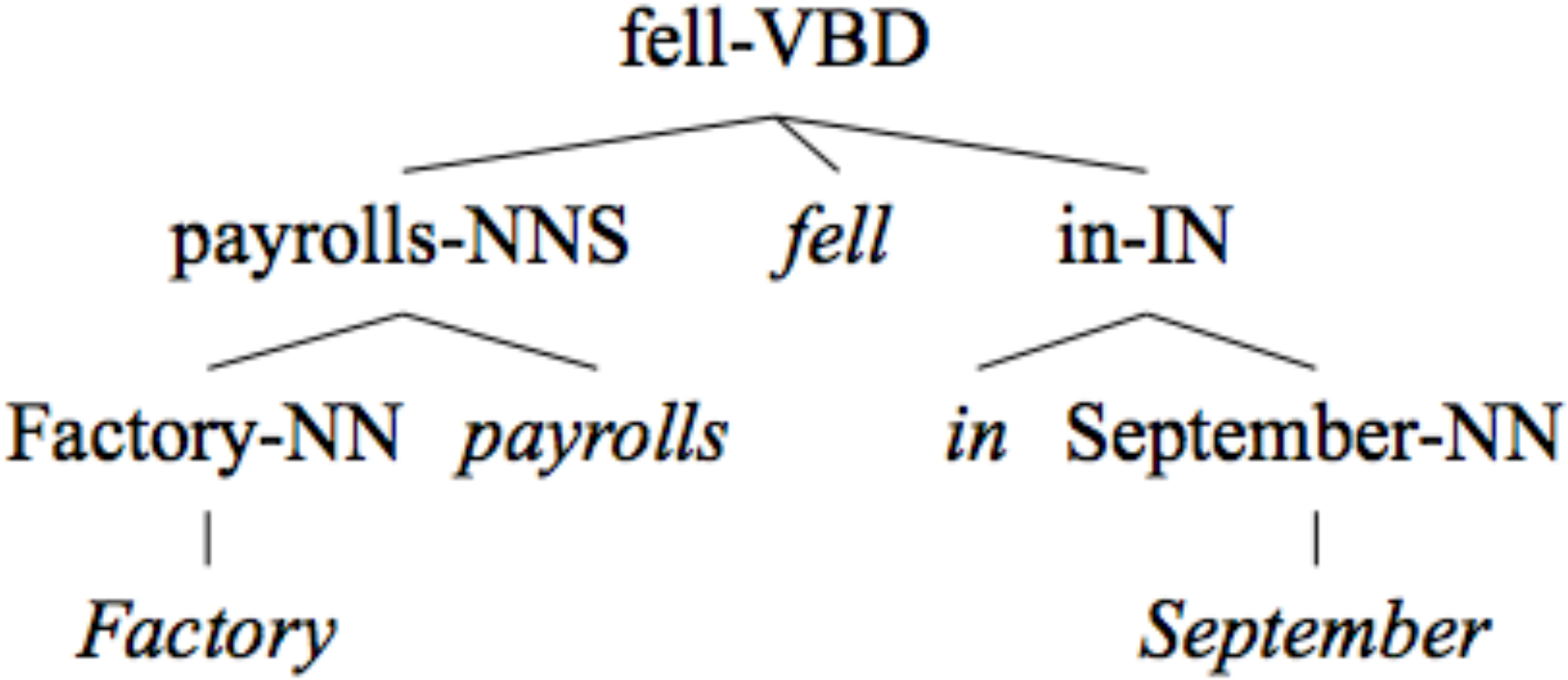


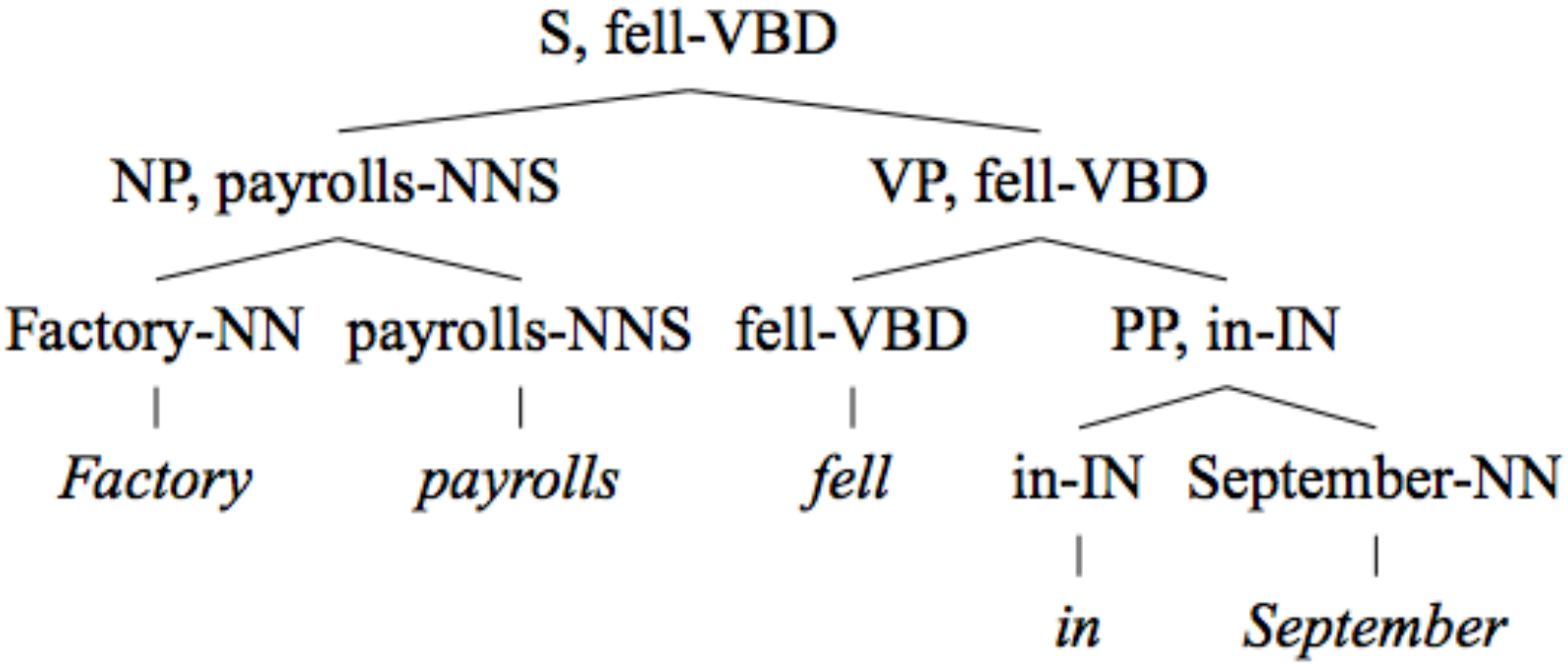
Figure 7: An error resolved with the SPLIT-VP annotation: (a) the incorrect baseline parse and (b) the correct SPLIT-VP parse.

Make a dependency parser, also



Combine them into the same parser

- P(parse) ∝ P(PCFG) * P(dependency)
- P(parse) = 0 if the structures are incompatible



Using the Stanford parser with NLTK?

- It's a Java program
  - Probably run it as a separate process
- It gives its output in the standard tree format
- **nltk.Tree.parse** can turn it into an NLTK tree