CS 378 Lecture 13

Today
- Finish constituency parsing:
  - Evaluation
  - Better PCFGs
- Dependencies
- (start) Shift-reduce parsing

Recap Building a parser:

Input: treebank

```
S
  NP
    PRP
    VBD
    DT
    NN
    IN
    PRP
  VP
    NP
    PP
  S
S
S
```

He gave the book to her
1. Binarize:

\[
S \rightarrow NP \rightarrow VP
\]

\[
VP \rightarrow VBD \rightarrow VP[NP PP]
\]

\[
NP \rightarrow PP
\]

2. Count and normalize to get PCFG probs:

**NP:**

\[
P(NP \rightarrow PRP | NP) = \frac{1}{2}
\]

\[
P(NP \rightarrow DT \ NN | NP) = \frac{1}{2}
\]

Do this over all trees.

3. CKY: find \( \arg \max_T P(T | \bar{x}) \)

\[
= \arg \max_T P(T, \bar{x})
\]
Announcements
- A3 updated Q3b
- Final proj: independent proposal deadline pushed back (Oct 27)
- Midterm: list of topics posted
- Next class: review

Finishing CKY

① Runtime: n words in a sentence
\[ O(n^3) \]
Chart has \( O(n^2) \) cells
Each cell: loop over \( O(n) \) values of \( K \)

② Unaries:

\[ S \rightarrow NP \]
**Shift-reduce parsing**

Left-to-right

Parse in linear time

States partially built parse on a stack

Move through the sentence, add words to the partial parse as we go

Greedy, approximate