Stochastic Parsing

Scott Farrar
CLMA, University of Washington
farrar@u.washington.edu

January 20, 2010
Today’s lecture

1. Probabilistic parsing
   - Probabilistic CKY

2. Homework 3
Deterministic parsing

Using deterministic methods, it’s difficult to tell which parses are correct.
Deterministic parsing

Using deterministic methods, it’s difficult to tell which parses are correct.

**Strategy:** prune search space by eliminating suboptimal or improbable ones
Deterministic parsing

Using deterministic methods, it’s difficult to tell which parses are correct.

**Strategy**: prune search space by eliminating suboptimal or improbable ones

Use a PCFG to guide the pruning process; chose the best parse, or $n$ best parses.
CKY vs. Prob-CKY

In the non-probabilistic version, what's contained in a CKY cell?

All possible structures for a given span of input; in other words, all possible syntactic interpretations for a given substring.
CKY vs. Prob-CKY

- In the non-probabilistic version, what’s contained in a CKY cell?
- All possible structures for a given span of input; in other words, all possible syntactic interpretations for a given substring.

...(time flies)... can be a VP, S, NP, ...

CKY vs. Prob-CKY

- In the non-probabilistic version, what’s contained in a CKY cell?
- All possible structures for a given span of input; in other words, all possible syntactic interpretations for a given substring.
  \( \ldots (time\ flies) \ldots \) can be a VP, S, NP, ...
- What if we only need the most likely parse (or top 10 most likely parses)?
function Probabilistic-CKY(words, grammar) returns most probable parse, probability
  for j ← from 1 to LENGTH(words) do
    for all \( \{A|A \rightarrow \text{words}[j] \in \text{grammar} \} \)
      \( \text{table}[j - 1, j, A] \leftarrow P(A \rightarrow \text{words}[j]) \)
    for i ← from \( j - 2 \) downto 0 do
      for k ← i + 1 to \( j - 1 \) do
        for all \( \{A|A \rightarrow B \ C \in \text{grammar}, \) and \( \text{table}[i, k, B] > 0 \) and \( \text{table}[k, j, C] > 0 \} \)
          if \( \text{table}[i, j, A] < P(A \rightarrow B \ C) \times \text{table}[i, k, B] \times \text{table}[k, j, C] \) then
            \( \text{table}[i, j, A] \leftarrow P(A \rightarrow B \ C) \times \text{table}[i, k, B] \times \text{table}[k, j, C] \)
        \( \text{back}[i, j, A] \leftarrow \{k, B, C\} \)
  return build_tree(back[1,LENGTH(words), S]), table[1,LENGTH(words), S]
See pcky_eg.pdf.
Today’s lecture

1. Probabilistic parsing
   - Probabilistic CKY

2. Homework 3
Homework 3

1. work with a real PCFG
2. build a probabilistic parser (CKY)
3. evaluate the results
Homework 3

1. work with a real PCFG
2. build a probabilistic parser (CKY)
3. evaluate the results

\[ P(\text{Hw3 is easy.}) = 0.0000001 \]
\[ P(\text{Hw3 is hard.}) = 0.004 \]
Parsing: dev/train/test paradigm

The Wall Street Journal (WSJ) section of the Penn Treebank (PTB), for all its faults, provides a very useful resource for comparing parser performance.
Parsing: dev/train/test paradigm

The Wall Street Journal (WSJ) section of the Penn Treebank (PTB), for all its faults, provides a very useful resource for comparing parser performance.

In building a probabilistic parser, there are four kinds of resources that are commonly used esp. in the ACL related literature:
The Wall Street Journal (WSJ) section of the Penn Treebank (PTB), for all its faults, provides a very useful resource for comparing parser performance.

In building a probabilistic parser, there are four kinds of resources that are commonly used esp. in the ACL related literature:

1. **training data**: large number of annotated sentences (sec. 2–21 of PTB has 39,830 sentences)
The Wall Street Journal (WSJ) section of the Penn Treebank (PTB), for all its faults, provides a very useful resource for comparing parser performance.

In building a probabilistic parser, there are four kinds of resources that are commonly used esp. in the ACL related literature:

1. **training data**: large number of annotated sentences (sec. 2–21 of PTB has 39,830 sentences)
2. **development data**: small number of annotated sentences used to “tweak” parser (sec. 22, of PTB)
The Wall Street Journal (WSJ) section of the Penn Treebank (PTB), for all its faults, provides a very useful resource for comparing parser performance.

In building a probabilistic parser, there are four kinds of resources that are commonly used esp. in the ACL related literature:

1. **training data**: large number of annotated sentences (sec. 2–21 of PTB has 39,830 sentences)
2. **development data**: small number of annotated sentences used to “tweak” parser (sec. 22, of PTB)
3. **test data**: small-medium number of un-annotated sentences used as input to parser (sec. 23 of PTB has 2416 sentences, ~ 6% of training set)
Parsing: dev/train/test paradigm

The Wall Street Journal (WSJ) section of the Penn Treebank (PTB), for all its faults, provides a very useful resource for comparing parser performance.

In building a probabilistic parser, there are four kinds of resources that are commonly used esp. in the ACL related literature:

1. **training data**: large number of annotated sentences (sec. 2–21 of PTB has 39,830 sentences)
2. **development data**: small number of annotated sentences used to “tweak” parser (sec. 22, of PTB)
3. **test data**: small-medium number of un-annotated sentences used as input to parser (sec. 23 of PTB has 2416 sentences, ~ 6% of training set)
4. **gold standard**: annotated version of test data, with no errors (hidden till parser is developed)