Logistics

• Assignment 2 is released. Due on November 14th, 11:59pm on blackboard.

• Start early!

• The proposal comments are on blackboard.

• Regrading: see corresponding TA (see the signature)

• Further comments: talk to the TA or the instructor

The grammar

\[
\begin{array}{c|c|c|c}
S & NP & VP & 0.9 \\
S & VP & 0.1 \\
VP & \rightarrow V & NP & 0.5 \\
VP & \rightarrow V & VP & 0.1 \\
NP & \rightarrow NP & V & 0.1 \\
NP & \rightarrow NP & V & 0.1 \\
VP & \rightarrow & V & 0.1 \\
NP & \rightarrow & NP & 0.7 \\
PP & \rightarrow & P & NP & 1.0 \\
N & \rightarrow & people & 0.5 \\
N & \rightarrow & fish & 0.2 \\
N & \rightarrow & tanks & 0.2 \\
N & \rightarrow & rods & 0.1 \\
V & \rightarrow & people & 0.1 \\
V & \rightarrow & fish & 0.6 \\
V & \rightarrow & tanks & 0.3 \\
P & \rightarrow & with & 1.0 \\
\end{array}
\]
### Extended CKY parsing

- CKY parsing is usually done after binarization
- Unaries can be incorporated into the algorithm
- Empties can be incorporated
- Doesn’t increase algorithmic complexity
- Binarization is vital
- Without binarization, you don’t get parsing cubic in the length of the sentence and in the number of nonterminals in the grammar

### Treebanks

- **English Penn Treebank**: Standard corpus for testing syntactic parsing consists of 1.2 M words of text from the Wall Street Journal (WSJ). Typical to train on about 40,000 parsed sentences and test on an additional standard disjoint test set of 2,416 sentences.
- **Chinese Penn Treebank**: 100K words from the Xinhua news service.
- Other corpora existing in many languages, see the Wikipedia article “Treebank”
Evaluating constituency parsing

Gold standard brackets:
S (0:11), NP (0:2), VP (2:9), NP (4:6), PP (6-9), NP (7:10)

Candidate brackets:
S (0:11), NP (0:2), VP (2:9), NP (4:6), PP (6-9), NP (7:10)

Labeled Precision: 3/7 = 42.9%
Labeled Recall: 3/8 = 37.5%
F1: 40.0%
POS Tagging Accuracy: 11/11 = 100.0%

How good are PCFGs?

• Penn WSJ parsing accuracy: about 73% F1 with basic classifiers
  (now state-of-the-art neural models can reach 91-92% F1)
• Robust
  • Usually admit everything, but with low probability
• Partial solution for grammar ambiguity
  • A PCFG gives some idea of the plausibility of a parse
  • Give a probabilistic language model
  • But in the simple case it performs worse than a trigram model
• The problem seems to be that PCFGs lack the lexicalization of a trigram model (i.e. considering the words themselves)

(Head) Lexicalization of PCFGs

The head word of a phrase gives a good representation of the phrase’s structure and meaning (head words are decided by rules)

• Puts the properties of words back into a PCFG

Head Words

• Syntactic phrases usually have a word in them that is most “central” to the phrase.
• Linguists have defined the concept of a lexical head of a phrase.
• Simple rules can identify the head of any phrase by percolating head words up the parse tree.
  • Head of a VP is the main verb
  • Head of an NP is the main noun
  • Head of a PP is the preposition
  • Head of a sentence is the head of its VP

(Head) Lexicalization of PCFGs
Lexicalization of PCFGs: Charniak (1997)

- A very straightforward model of a lexicalized PCFG
- Probabilistic conditioning is “top-down” like a regular PCFG
- But actual parsing is bottom-up, somewhat like the CKY algorithm we saw

Lexicalization models argument selection by sharpening rule expansion probabilities

- The probability of different verbal complement frames (i.e., “subcategorizations”) depends on the verb.

Human Parsing

- Computational parsers can be used to predict human reading time as measured by tracking the time taken to read each word in a sentence.
- Psycholinguistic studies show that words that are more probable given the preceding lexical and syntactic context are read faster.
  - John put the dog in the pen with a lock.
  - John put the dog in the pen with a bone in the car.
  - John put the dog in the pen with a bone.
- Modeling these effects requires an incremental statistical parser that incorporates one word at a time into a continuously growing parse tree.
Garden Path Sentences
- People are confused by sentences that seem to have a particular syntactic structure but then suddenly violate this structure, so the listener is "lead down the garden path".
- The complex houses married students.
- The old man the sea.
- While Anna dressed the baby spit up on the bed.
- Incremental computational parsers can try to predict and explain the problems encountered parsing such sentences.

Center Embedding
- Nested expressions are hard for humans to process beyond 1 or 2 levels of nesting.
- The rat the cat the dog bit chased died.
- The rat the cat the dog the boy owned bit chased died.
- Requires remembering and popping incomplete constituents from a stack and strains human short-term memory.
- Equivalent "tail embedded" (tail recursive) versions are easier to understand since no stack is required.
- The boy owned a dog that bit a cat that chased a rat that died.

Dependency Grammar and Dependency Structure
- Dependency syntax postulates that syntactic structure consists of lexical items linked by binary asymmetric relations ("arrows") called dependencies.
- The arrows are commonly typed with the name of grammatical relations (subject, prepositional object, apposition, etc.)
- Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas.

Relation between phrase structure and dependency structure
- A dependency grammar has a notion of a head. Officially, CFGs don’t.
- But modern linguistic theory and all modern statistical parsers (Charniak, Collins, Stanford, …) do, via hand-written phrasal "head rules":
  - The head of a Noun Phrase is a noun/nounphrase/adj/…
  - The head of a Verb Phrase is a verb/modifier/infinitive/adj/…
  - The head rules can be used to extract a dependency parse from a CFG parse.

Dependency Graph from Parse Tree
- Can convert a phrase structure parse to a dependency tree by making the head of each non-head child of a node depend on the head of the head child.
Methods of Dependency Parsing

1. Dynamic programming (like in the CKY algorithm)
   You can do it similarly to lexicalized PCFG parsing: an \(O(n^3)\) algorithm
   Eisner (1996) gives a clever algorithm that reduces the complexity to \(O(n^2)\), by
   producing parse items with heads at the ends rather than in the middle

2. Graph algorithms
   You create a Maximum Spanning Tree for a sentence
   McDonald et al.'s (2005) MSTParser scores dependencies independently using a ML
   classifier (he uses MIRA, for online learning, but it could be MaxEnt)

3. Constraint Satisfaction
   Edges are eliminated that don’t satisfy hard constraints. Karlsson (1990), etc.

4. "Deterministic parsing"
   Greedy choice of attachments guided by machine learning classifiers
   MaltParser (Nivre et al. 2008) – discussed in the next segment