ACS Introduction to NLP

Lecture 5: PCFG Models for Statistical Parsing

UNIVERSITY OF CAMBRIDGE

Stephen Clark

Natural Language and Information Processing (NLIP) Group

sc609@cam.ac.uk

Interesting Ambiguity Examples

- *The a are of I*
- *The cows are grazing in the meadow*
- *John saw Mary*

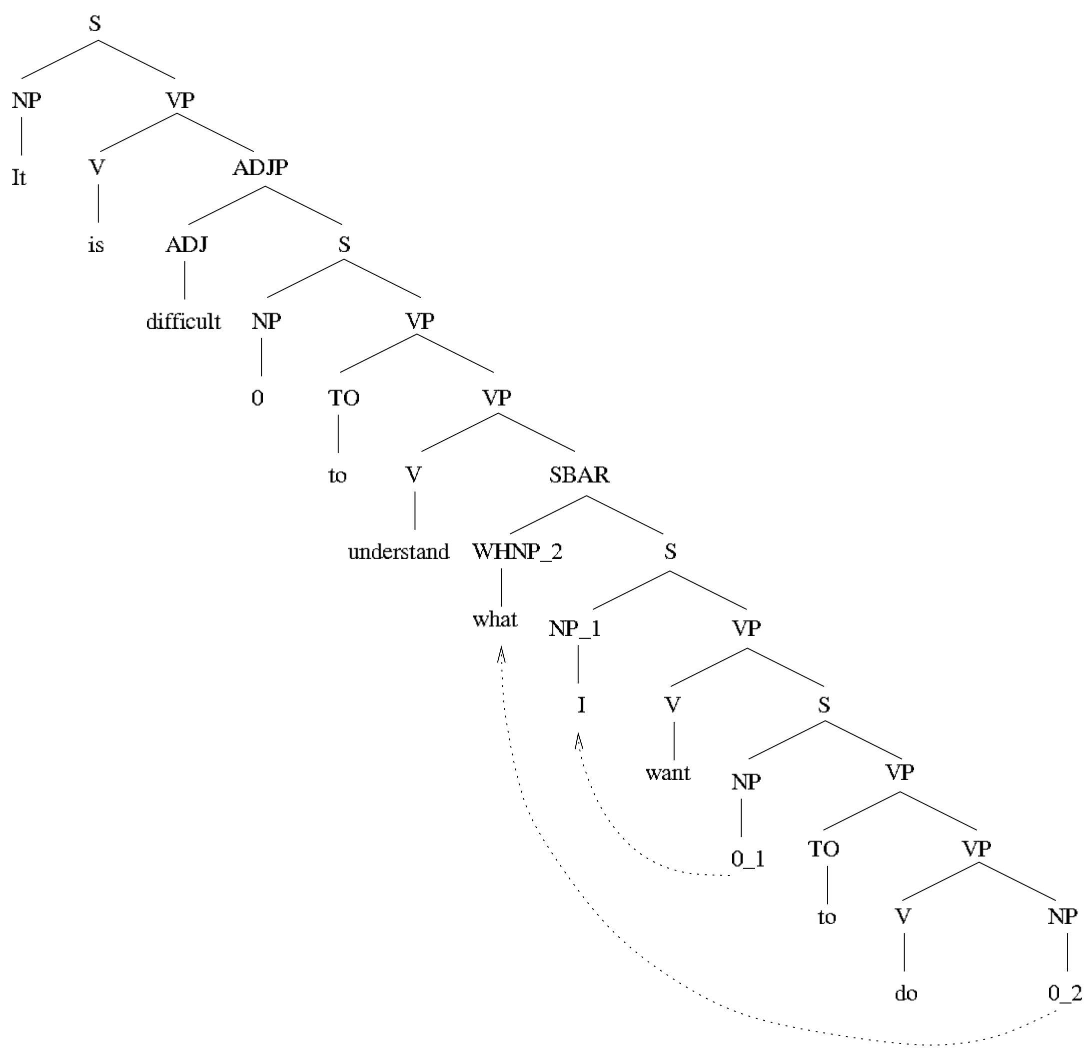
examples from Abney (1996)

The Penn Treebank

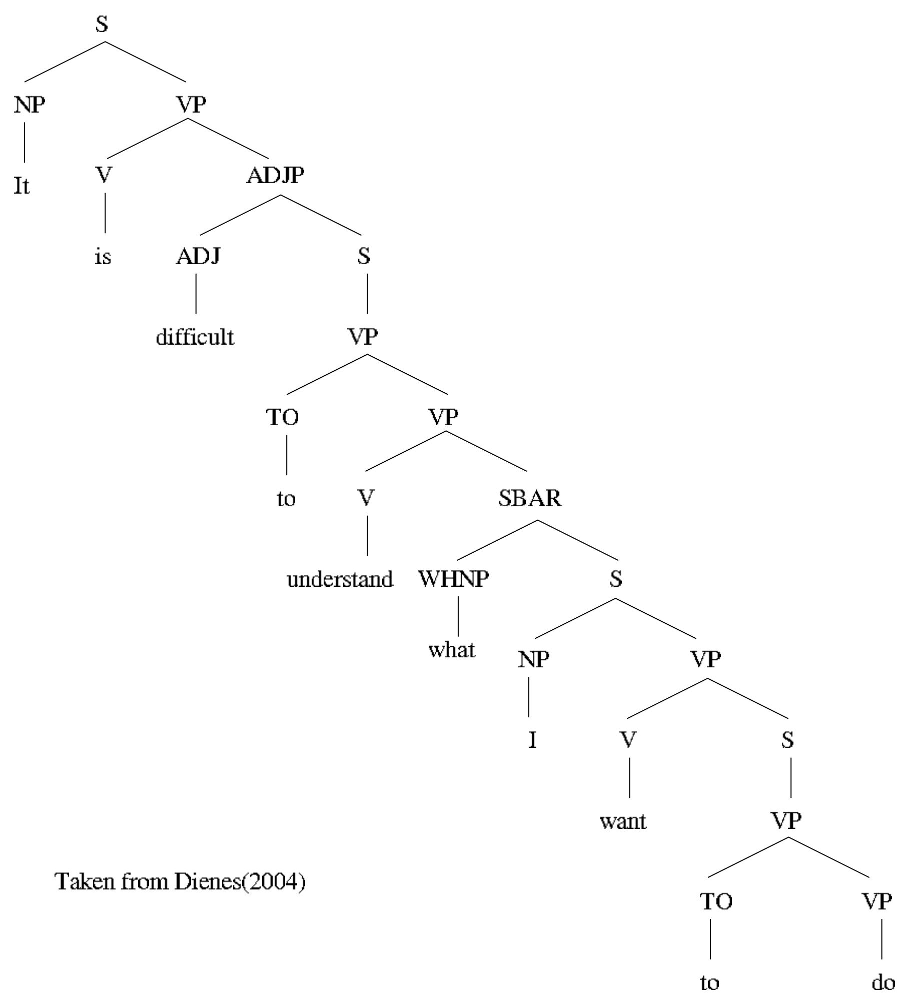
- 40,000 WSJ newspaper sentences annotated with phrase-structure trees
- The trees contain some predicate-argument information and traces
- Created in the early 90s
- Produced by automatically parsing the newspaper sentences followed by manual correction
- Took around 3 years to create
- Sparked a parsing “competition” which is still running today
  - leading some commentators to describe the last 15 years of NLP as the study of the WSJ

[comment on methodology]

An Example Penn Treebank Tree



A Tree a typical PTB Parser would produce



Taken from Dienes(2004)

Characterisation of Statistical Parsing I

- What is the grammar which determines the set of legal syntactic structures for a sentence? How is that grammar obtained?
- What is the algorithm for determining the set of legal parses for a sentence (given a grammar)?
- What is the model for determining the plausibility of different parses for a sentence?
- What is the algorithm, given the model and a set of possible parses, which finds the best parse?

Characterisation of Statistical Parsing II

$$T_{\text{best}} = \arg\max_{T} \textit{Score}(T, S)$$

- Just two components:
  - the *model*: a function *Score* which assigns scores (probabilities) to tree, sentence pairs
  - the *parser*: the algorithm which implements the search for T_{best}
- Statistical parsing seen as more of a pattern recognition/Machine Learning problem plus search
  - the grammar is only implicitly defined by the training data and the method used by the parser for generating hypotheses

Statistical Parsing Models

- Probabilistic approach would suggest the following *Score* function:

$$\textit{Score}(T, S) = P(T|S)$$

- Lots of research on different probability models for Penn Treebank trees
  - generative models, log-linear (maxent) models, perceptron, . . .

Generative Models

$$\begin{aligned}\arg\max_T P(T|S) &= \arg\max_T \frac{P(T,S)}{P(S)} \\ &= \arg\max_T P(T,S)\end{aligned}$$

- Why model the joint probability when the sentence is given?
- Modelling a parse as a generative process allows the parse to be broken into manageable parts, for which the corresponding probabilities can be reliably estimated
- Probability estimation is easy for these sorts of models (ignoring smoothing issues)
  - maximum likelihood estimation = relative frequency estimation
- But choosing how to break up the parse is something of a black art

Probabilistic Context Free Grammars (PCFGs)

- A PCFG is a CFG with a set of probability distributions on the rules:

$$\sum_{\alpha} P(X \rightarrow \alpha) = 1$$

- Simple example (generating some ungrammatical sentences):

$S \rightarrow NP \quad VP \quad 1.0$
$VP \rightarrow V \quad 0.1$
$VP \rightarrow V \quad NP \quad 0.7$
$VP \rightarrow V \quad NP \quad NP \quad 0.2$
$NP \rightarrow Det \quad N \quad 0.6$
$NP \rightarrow N \quad 0.4$
$Det \rightarrow the \quad 0.5$
$Det \rightarrow a \quad 0.5$
$N \rightarrow cat \quad 0.2$
$N \rightarrow dog \quad 0.2$
$N \rightarrow man \quad 0.3$
$N \rightarrow woman \quad 0.3$
$V \rightarrow chased \quad 0.8$
$V \rightarrow kissed \quad 0.2$

Probability of a Tree

- Joint probability of a tree T and sentence S is just the product of the probabilities of the rules used to build the tree
- For example, the probability of the tree associated with *the cat chased a dog*, using the previous grammar, is as follows:

$$P(S) \times P(S \rightarrow NP\ VP|S) \times P(NP \rightarrow Det\ N|NP) \times P(VP \rightarrow V\ NP|VP) \times$$
$$P(NP \rightarrow Det\ N|NP) \times P(Det \rightarrow the|Det) \times P(N \rightarrow cat|N) \times$$
$$P(V \rightarrow chased|V) \times P(Det \rightarrow a|Det) \times P(N \rightarrow dog|N)$$
$$= 1.0 \times 1.0 \times 0.6 \times 0.7 \times 0.6 \times 0.5 \times 0.2 \times 0.8 \times 0.5 \times 0.2$$

Probability of a Tree

- Think of a random “generative process” as having generated the tree top-down, according to the rule probabilities
- The probability above is just an application of the chain rule, plus independence assumptions (similar to the HMM for the tagging case)
- Independence assumption is that the probability of rewriting a non-terminal in a particular way only depends on the non-terminal, and nothing else in the tree
- Similar idea to the notion of context-freeness in the non-probabilistic grammar case

Obtaining the Grammar

- A CFG can be read directly off the trees in the PTB
- For the tree on p.4, for example, we would get rules such as:

$$S \longrightarrow NP\ \ VP$$
$$NP \longrightarrow It$$
$$VP \longrightarrow V\ \ ADJP$$
$$S \longrightarrow VP$$

- Estimating the probabilities is easy!

$$\hat{P}(S \longrightarrow NP\ \ VP|S) = \mathit{freq}(S \longrightarrow NP\ \ VP)/\mathit{freq}(S)$$

- And relative frequency estimates are maximum likelihood estimates in this case (as they were for the HMM tagging model)

Problems with PCFGs

- The main problem is that a PCFG only has *structural* probabilities
- The words only have an effect at the leaves of the tree (by which time almost all of the tree has been generated)
- Consider trying to distinguish the parses for *John ate the pizza with a fork* and *John ate the pizza with the anchovies* using a PCFG

References

- Steven Abney (1996), Statistical Methods and Linguistics, available from Abney's webpage
- Chapter 11 of Manning and Schuetze
- Michael Collins (1999), Head-Driven Statistical Models for Natural Language Parsing, UPenn PhD thesis