Probabilistic Context-Free Grammars

Informatics 2A: Lecture 19

Mirella Lapata

School of Informatics
University of Edinburgh

01 November 2011

Reading:

J&M 2^{nd} edition, ch. 14 (Introduction → Section 14.2)

Motivation

Three things motivate the use of probabilities in grammars and parsing:

1. Syntactic disambiguation – main motivation
2. Coverage – issues in developing a grammar for a language
3. Representativeness – adapting a parser to new domains, texts.

Motivation 1: Ambiguity

- Amount of ambiguity increases with sentence length.
- Real sentences are fairly long (avg. sentence length in the *Wall Street Journal* is 25 words).
- The amount of (unexpected!) ambiguity increases rapidly with sentence length. This poses a problem, even for chart parsers, if they have to keep track of all possible analyses.
- It would reduce the amount of work required if we could ignore improbable analyses.

> A second provision passed by the Senate and House would eliminate a rule allowing companies that post losses resulting from LBO debt to receive refunds of taxes paid over the previous three years. [wsj_1822] (33 words)

Motivation 2: Coverage

- It is actually very difficult to write a grammar that covers all the constructions used in ordinary text or speech.
- Typically hundreds of rules are required in order to capture both all the different linguistic patterns and all the different possible analyses of the same pattern. (How many grammar rules did we have to add to cover three different analyses of *You made her duck*?)
- Ideally, one wants to induce (learn) a grammar from a corpus.
- Grammar induction requires probabilities.

Motivation 3: Representativeness

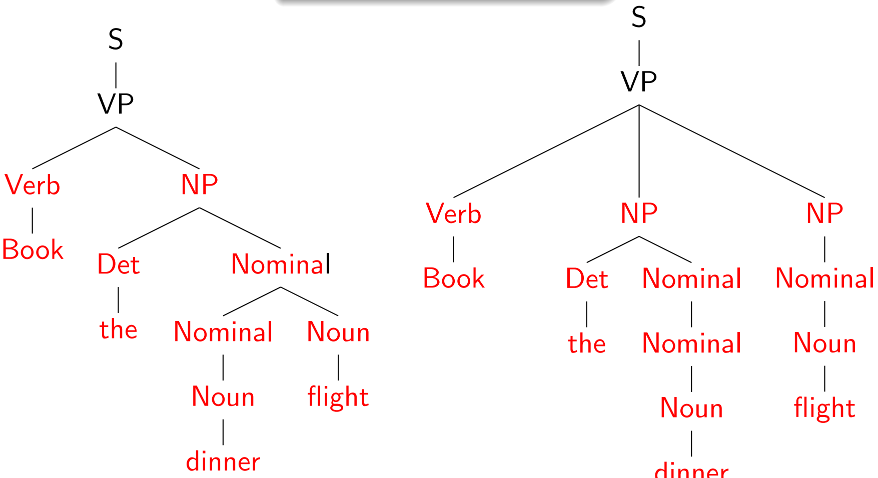
The likelihood of a particular construction can vary, depending on:

- register (formal vs. informal): eg, *greenish*, *alot*, subject-drop (*Want a beer?*) are all more probable in informal than formal register;
- genre (newspapers, essays, mystery stories, jokes, ads, etc.): Clear from the difference in PoS-taggers trained on different genres in the Brown Corpus.
- domain (biology, patent law, football, etc.).

Probabilistic grammars and parsers can reflect these kinds of distributions.

Example Parses for an Ambiguous Sentence

Book the dinner flight.



Probabilistic Context-Free Grammars

A PCFG $\langle N, \Sigma, R, S\rangle$ is defined as follows:

N	is the set of non-terminal symbols
Σ	is the terminals (disjoint from N)
R	is a set of rules of the form $A \rightarrow \beta[p]$ where $A \in N$ and $\beta \in (\sigma \cup N)*$, and p is a number between 0 and 1
S	a start symbol, $S \in N$

A PCFG is a CFG in which each rule is associated with a probability.

More about PCFGS

What does the p associated with each rule express?

It expresses the probability that the LHS non-terminal will be expanded as the RHS sequence.

- $P(A \rightarrow \beta | A)$
- $\sum_{\beta} P(A \rightarrow \beta | A) = 1$
- The sum of the probabilities associated with all of the rules expanding the non-terminal A is 1

$$A \rightarrow \beta \; [p] \quad \text{or} \quad P(A \rightarrow \beta | A) = p \quad \text{or} \quad P(A \rightarrow \beta) = p$$

Example Grammar

$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow the$	[.10]
$S \rightarrow Aux\ NP\ VP$	[.15]	$Det \rightarrow a$	[.90]
$S \rightarrow VP$	[.05]	$Noun \rightarrow book$	[.10]
$NP \rightarrow Pronoun$	[.35]	$Noun \rightarrow flight$	[.30]
$NP \rightarrow Proper\text{-}Noun$	[.30]	$Noun \rightarrow dinner$	[.60]
$NP \rightarrow Det\ Nominal$	[.15]	$Proper\text{-}Noun \rightarrow Houston$	[.60]
$NP \rightarrow Nominal$	[.15]	$Proper\text{-}Noun \rightarrow NWA$	[.40]
$Nominal \rightarrow Noun$	[.75]	$Aux \rightarrow does$	[.60]
$Nominal \rightarrow Nominal\ Noun$	[.05]	$Aux \rightarrow can$	[.40]
$VP \rightarrow Verb$	[.35]	$Verb \rightarrow book$	[.30]
$VP \rightarrow Verb\ NP$	[.20]	$Verb \rightarrow include$	[.30]
$VP \rightarrow Verb\ NP\ PP$	[.10]	$Verb \rightarrow prefer$	[.20]
$VP \rightarrow Verb\ PP$	[.15]	$Verb \rightarrow sleep$	[.20]

PCFGs and disambiguation

- A PCFG assigns a probability to every parse tree or derivation associated with a sentence.
- This probability is the product of the rules applied in building the parse tree.

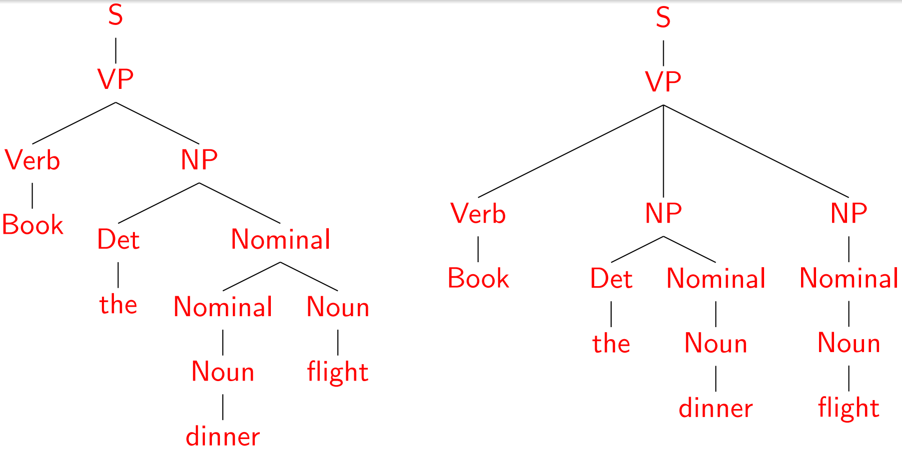
$$P(T,S) = \prod_{i=1}^{n} P(A_i \rightarrow \beta_i) \; n \; is \; number \; of \; rules \; in \; T$$

$$P(T,S) = P(T)P(S|T) = P(S)P(T|S) \; by \; definition$$

$$But \; P(S|T) = 1 \quad because \; all \; the \; words \; in \; S \; are \; in \; T$$

$$So, P(T,S) = P(T)$$

Application 1: Disambiguation



$$P(T_{left}) = .05 * .20 * .20 * .20 * .75 * .30 * .60 * .10 * .40 = \mathbf{2.2} \times \mathbf{10^{-6}}$$
$$P(T_{right}) = .05 * .10 * .20 * .15 * .75 * .75 * .30 * .60 * .10 * .40 = \mathbf{6.1} \times \mathbf{10^{-7}}$$

Application 2: Language Modeling

As well as assigning probabilities to parse trees, a PCFG assigns a probability to every sentence generated by the grammar. This is useful for language modeling.

The probability of a sentence is the sum of the probabilities of each parse tree associated with the sentence:

$$P(S) = \sum_{T s.t. yield(T)=S} P(T, S)$$

$$P(S) = \sum_{s.t. yield(T)=S} P(T)$$

When is it useful to know the probability of a sentence?
When ranking the output of speech recognition, machine translation, and error correction systems.

Probabilistic CKY

Many probabilistic parsers use a probabilistic version of the CKY bottom-up chart parsing algorithm.

Sentence S of length n and CFG grammar with V non-terminals

Normal CKY

2-$d(n+1)*(n+1)$ array where a value in cell (i,j) is list of non-terminals spanning position i through j in S.

Probabilistic CKY

3-$d(n+1)*(n+1)*V$ array where a value in cell (i,j,K) is probability of non-terminal K spanning position i through j in S

As with regular CKY, probabilistic CKY assumes that the grammar is in Chomsky-normal form (rules $A \rightarrow B\ C$ or $A \rightarrow w$).

Probabilistic CKY

function Probabilistic-CKY(*words*, *grammar*) **returns** most probable parse and its probability

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A | A \rightarrow words[j] \in grammar\}$
   $table[j-1, j, A] \leftarrow P(A \rightarrow words[j])$
  for $i \leftarrow$ **from** $j - 2$ **downto** 0 **do**
    for all $\{A | A \rightarrow BC \in grammar,$
        and $table[i, k, B] > 0$ **and** $table[k, j, C] > 0\}$
     if$(table[i, j, A] < P(A \rightarrow BC) \times table[i, k, B] \times table[k, j, C])$**then**
      $table[i, j, A] \leftarrow P(A \rightarrow BC) \times table[i, k, B] \times table[k, j, C]$
      $back[i, j, A] \leftarrow \{k, B, C\}$

return
BUILD_TREE(*back*[1,LENGTH(*words*), *S*]), *table*[1,LENGTH(*words*), *S*]

Visualizing the Chart

The	flight	includes	a	meal
Det: .40 [0, 1]	NP: $.30 \times .40 \times .02 = .0024$ [0, 2]	[0, 3]	[0, 4]	S: $.80 \times .0024 \times .000012 = .000000023$ [0, 5]
	N: .02 [1, 2]	[1, 3]	[1, 4]	[1, 5]
		V: .05 [2, 3]	[2, 4]	VP: $.20 \times .05 \times 0.0012 = 0.000012$ [2, 5]
			Det: .40 [3, 4]	NP: $.30 \times .40 \times .01 = 0.0012$ [3, 5]
				N: .01 [4, 5]

$S \rightarrow NP\ VP$ .80	$Det \rightarrow the$ .40
$NP \rightarrow Det\ N$ .30	$Det \rightarrow a$ .40
$VP \rightarrow V\ NP$ .20	$N \rightarrow meal$ .01
$V \rightarrow includes$ .05	$N \rightarrow flight$ .02

Clicker Questions

$S \longrightarrow NP\ VP$	$Det \longrightarrow the$
$NP \longrightarrow Det\ N$	$Det \longrightarrow a$
$VP \longrightarrow V\ NP$	$N \longrightarrow meal$
$V \longrightarrow includes$	$N \longrightarrow flight$

1. Assume someone tells you that the rules of the grammar above are equally likely. What is the probability of $S \longrightarrow NP\ VP$?

   (a) 1 (b) 0.5 (c) $\frac{1}{8}$ (d) 2

2. How does HMM tagging relate to PCFGs?

   (a) It really doesn't, they are both probabilistic.
   (b) It could be used to obtain the terminal probabilities.
   (c) HMM tagging also uses CYK.

Summary

- A PCFG is a CFG with each rule annotated with a probability;
- the sum of the probabilities of all rules that expand the same non-terminal must be 1;
- probability of a parse tree is the product of the probabilities of all the rules used in this parse;
- probability of sentence is sum of probabilities of all its parses;
- applications for PCFGs: disambiguation, language modeling;
- Probabilistic CKY algorithm.

Next lecture: But where do the rule probabilities come from?