CMPT 825: Natural Language Processing **Spring 2006**

Lecture 11 — Feb 23, 2006

Lecturer: Anoop Sarkar *Scribe: Mehdi M. Kashani*

Probabilistic Parsers

The difference between Natural Language Processing and Computational Linguistics: In NLP it is allowed to apply any method whether the model makes sense or not. In this case, the right answer matters. In CL, it is tried to find a generative story and understand the nature of the model.

Natural Language Understanding:
Speech/text $\overset{map}{\longleftrightarrow}$ Meaning
French $\overset{map}{\rightarrow}$ English

Example: John saw the man with a telescope.

Standard model of Natural Language Understanding:

M1 $\rightarrow$ Formula1 $\rightarrow$ (compatible with) Model1
M2 $\rightarrow$ Formula2 $\rightarrow$ (compatible with) Model2

Formulas can be predicates in logic or lambda function, or even a picture.

$$\lambda X \lambda Y see(X, Y) - - > \lambda Y see(John, Y), with(man, telescope) \tag{11.1}$$

or with(see(John,man),telescope)

There is a current model that is true.
There are different representations of a model. Other than the tree shown in Charniak's paper, one might use Dependency Grammar or Categorical Grammar.
There can be a huge number of possible parse trees for a given sentence. In Statistical Parsing we only compute plausible ones (higher ranks). In this case knowledge acquisition becomes important.

Pre-terminal is the same thing as part of speech tag.
Training is performed on the labeled data. For example, to compute the probability of a noun given the determiner, the frequency of happening determiner and noun is divided on the frequency of determiner:

$$P(n|det) = \frac{f(det, n)}{f(det)} \tag{11.2}$$

$$t^* = argmax_t P(t|s) \tag{11.3}$$

$$p(t|s) = P(t) * P(s|t) = \Pi \underbrace{P(t_i|t_{i-2}, t_{i-1})}_{Trigram\ Model} * \underbrace{P(w_i|t_i)}_{Emission} \tag{11.4}$$

Example

The sentence: The can might explode.
By assuming independence:
P(TAGS|STRING) = P(<det,noun,md>|<the,can,might>) =
P(det|bos,bos)*P(the|det) *
P(noun|bos,det)*P(can|nn) *
P(md|det,noun)*P(might|md)

How to find probabilities in the rules:

$$A- > \alpha \tag{11.5}$$

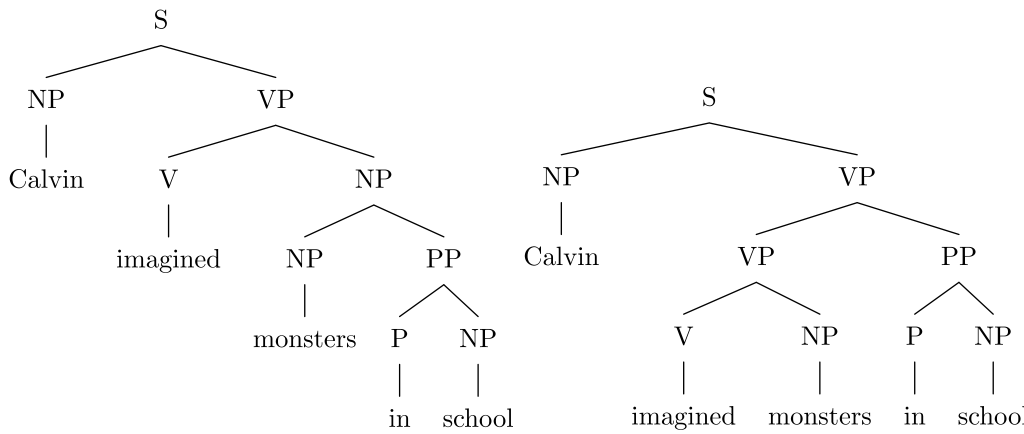
$$P = \frac{count(A- > \alpha)}{\sum_\alpha count(A- > \alpha)} \tag{11.6}$$

But does the model fit the problem? What happens for example if we increase the number of NP terminals?

Example

Calvin imagined monsters in school.
S → NP VP (P=1)
VP → V NP (P=0.9)
VP → VP PP (P=0.1)
PP → P NP (P=1)

NP → NP PP (P=0.25)
NP → Calvin | monster | school (P=0.25 each)
V → imagined
prep → in
Probabilities help us to decide which tree to choose.



To solve the PP attachment problem we can look into the training data. For example, considering the above instances and assuming that they are in the training data, We know each of them are attached to which part of speech:
imagined monsters in school. → attached to NP
bought shirt with pockets. → attached to NP
bought shirt with credit card. → attached to VP

The performance of different kinds of taggers:
81% Naive
84% Using Katz Back-off
87.5% Using Word Classes
88.1% Using Wordnet for word classes
88% Human by using 4 previous words.
94% Human by using the whole sentence.

$$\begin{array}{lcl} \text{Sound} & \longrightarrow & \text{Meaning} \\ \downarrow & & \downarrow \\ \text{Text} & & \text{Logic} \\ \downarrow & & \downarrow \end{array}$$

Sentence $\overset{PCFG}{\rightarrow}$ Tree
Sentence $\overset{PFSA}{\rightarrow}$ Part of Speech

Example

A program to provide safety in trucks and minivans.

In this example, one can have at least three different interpretations:

1. **trucks** and **minivans** are attached to **in**.
2. **safety in trucks** and **minivans** are attached to **provide**.
3. **a program to provide safety in trucks** and **minivans** are attached to the same thing (not in this sentence).

Y	X
1	join(v) board(n) as(p) director(n2)
0	...
0	...
1	...

Suppose we encode verb and noun, respectively as one and zero. So, if in the training data P(1—v,n,p,n2) ¿ 0.5 then we can infer it is attached to the verb and otherwise to the noun.
For example:

$$P(1|v,n,p,n2) = \frac{f(1,v,n,p,n2)}{f(v,n,p,n2)} \;\; if \; f(v,n,p,n2) > 0$$

$$Label\ Frequency: \; \frac{f(1)}{N} = 0.46 \; and \; \frac{f(0)}{N} = 0.54$$

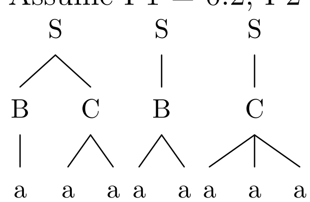
But in general smoothing should be performed. If Katz backoff is used:

$$
\begin{aligned}
P(1|v,n,p,n2) &= \frac{f(1,v,n,p,n2)}{f(v,n,p,n2)} \ \ if\ f(v,n,p,n2) > 0 \\
&= \frac{f(1,v,n,p) + f(1,v,p,n2) + f(1,n,p,n2)}{f(v,n,p) + f(v,p,n2) + f(n,p,n2)} \ \ if\ denominator > 0 \\
&= \frac{f(1,v,p) + f(1,n,p) + f(1,p,n2)}{f(v,p) + f(n,p) + f(p,n2)} \ \ if denominator > 0 \\
&= \frac{f(1,p)}{f(p)} \ \ if f(p) > 0 \\
&= 0
\end{aligned}
$$

In all the cases of backing off p is kept as the head of PP. How do we decide it to be the head?
Linguistically speaking, it is because this is the right model. Statistically speaking you just have to look at the frequencies.

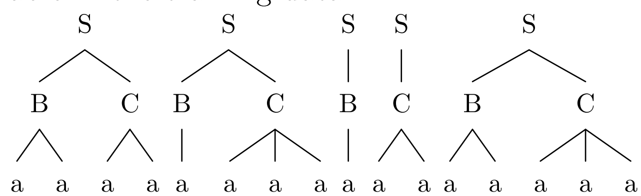
Small Quiz

Suppose in our training data we have the following parse trees, labeled as P1,P2 and P3. Find the probabilities of each of the CFG rules.
Assume P1 = 0.2, P2 = 0.1, P3 = 0.7

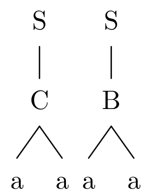


S → BC (P1=0.2)
S → B (P2=0.1)
S → C (P3=0.7)
B → a (P1/(P1+P2)=2/3)
B → aa (P2/(P1+P2)=1/3)
C → aa (P1/(P1+P3)=2/9)
C → aaa (P3/(P1+P3)=7/9)

Now, considering these rules, draw all the possible parse trees not available in the training data.



Next, for the input *aa* find the tree with the highest probability.
The trees that are found are the following:



Where the first one is the most probable. Notice that its probability in the training data is zero.