CMPT-413: Computational Linguistics

Anoop Sarkar
http://www.cs.sfu.ca/~anoop

February 28, 2013

Probabilistic CFG (PCFG)

$$
\begin{array}{cccc}
S & \rightarrow & NP\ VP & 1 \\
VP & \rightarrow & V\ NP & 0.9 \\
VP & \rightarrow & VP\ PP & 0.1 \\
PP & \rightarrow & P\ NP & 1 \\
NP & \rightarrow & NP\ PP & 0.25 \\
NP & \rightarrow & Calvin & 0.25 \\
NP & \rightarrow & monsters & 0.25 \\
NP & \rightarrow & school & 0.25 \\
V & \rightarrow & imagined & 1 \\
P & \rightarrow & in & 1
\end{array}
$$

$$P(input) = \sum_{tree} P(tree \mid input)$$

$$P(Calvin\ imagined\ monsters\ in\ school) = ?$$

Notice that $P(VP \rightarrow V\ NP) + P(VP \rightarrow VP\ PP) = 1.0$

Probabilistic CFG (PCFG)

$$P(\textit{Calvin imagined monsters in school}) = ?$$

```
(S (NP Calvin)
   (VP (V imagined)
       (NP (NP monsters)
           (PP (P in)
               (NP school)))))

(S (NP Calvin)
   (VP (VP (V imagined)
           (NP monsters))
       (PP (P in)
           (NP school))))
```

Probabilistic CFG (PCFG)

```
(S (NP Calvin)
   (VP (V imagined)
       (NP (NP monsters)
           (PP (P in)
               (NP school)))))
```

$$
\begin{aligned}
P(tree_1) &= P(S \rightarrow NP\ VP) \times P(NP \rightarrow Calvin) \times P(VP \rightarrow V\ NP) \times \\
&\quad P(V \rightarrow imagined) \times P(NP \rightarrow NP\ PP) \times P(NP \rightarrow monsters) \times \\
&\quad P(PP \rightarrow P\ NP) \times P(P \rightarrow in) \times P(NP \rightarrow school) \\
&= 1 \times 0.25 \times 0.9 \times 1 \times 0.25 \times 0.25 \times 1 \times 1 \times 0.25 = .003515625
\end{aligned}
$$

Probabilistic CFG (PCFG)

```
(S (NP Calvin)
   (VP (VP (V imagined)
           (NP monsters))
       (PP (P in)
           (NP school))))
```

$$
\begin{aligned}
P(tree_2) &= P(S \rightarrow NP\ VP) \times P(NP \rightarrow Calvin) \times P(VP \rightarrow VP\ PP) \times \\
&\quad P(VP \rightarrow V\ NP) \times P(V \rightarrow imagined) \times P(NP \rightarrow monsters) \times \\
&\quad P(PP \rightarrow P\ NP) \times P(P \rightarrow in) \times P(NP \rightarrow school) \\
&= 1 \times 0.25 \times 0.1 \times 0.9 \times 1 \times 0.25 \times 1 \times 1 \times 0.25 = .00140625
\end{aligned}
$$

Probabilistic CFG (PCFG)

$$
\begin{aligned}
P(\textit{Calvin imagined monsters in school}) &= P(\textit{tree}_1) + P(\textit{tree}_2) \\
&= .003515625 + .00140625 \\
&= .004921875 \\
\text{Most likely tree is } \textit{tree}_1 &= \underset{\textit{tree}}{\arg\max}\; P(\textit{tree} \mid \textit{input})
\end{aligned}
$$

```
(S (NP Calvin)
   (VP (V imagined)
       (NP (NP monsters)
           (PP (P in)
               (NP school)))))
```

```
(S (NP Calvin)
   (VP (VP (V imagined)
           (NP monsters))
       (PP (P in)
           (NP school))))
```

PCFG

- Central condition: $\sum_\alpha P(A \rightarrow \alpha) = 1$
- Called a *proper* PCFG if this condition holds
- Note that this means $P(A \rightarrow \alpha) = P(\alpha \mid A) = \frac{f(A,\alpha)}{f(A)}$
- $P(T \mid S) = \frac{P(T,S)}{P(S)} = P(T,S) = \prod_i P(RHS_i \mid LHS_i)$

PCFG

- What is the PCFG that can be extracted from this single tree:

```
(S    (NP (Det the) (NP man))
      (VP  (VP  (V played)
                (NP (Det a) (NP game)))
           (PP  (P with)
                (NP (Det the) (NP dog)))))
```

- How many different rhs α exist for $A \rightarrow \alpha$ where A can be *S, NP, VP, PP, Det, N, V, P*

PCFG

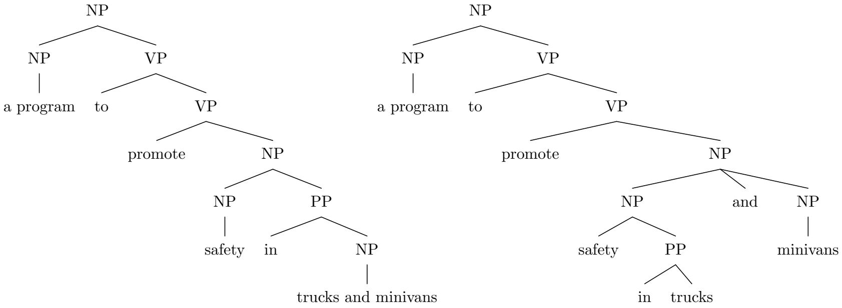
$$
\begin{array}{ccccll}
S & \rightarrow & NP\ VP & c = 1 & p = 1/1 & = 1.0 \\
NP & \rightarrow & Det\ NP & c = 3 & p = 3/6 & = 0.5 \\
NP & \rightarrow & man & c = 1 & p = 1/6 & = 0.1667 \\
NP & \rightarrow & game & c = 1 & p = 1/6 & = 0.1667 \\
NP & \rightarrow & dog & c = 1 & p = 1/6 & = 0.1667 \\
VP & \rightarrow & VP\ PP & c = 1 & p = 1/2 & = 0.5 \\
VP & \rightarrow & V\ NP & c = 1 & p = 1/2 & = 0.5 \\
PP & \rightarrow & P\ NP & c = 1 & p = 1/1 & = 1.0 \\
Det & \rightarrow & the & c = 2 & p = 2/3 & = 0.67 \\
Det & \rightarrow & a & c = 1 & p = 1/3 & = 0.33 \\
V & \rightarrow & played & c = 1 & p = 1/1 & = 1.0 \\
P & \rightarrow & with & c = 1 & p = 1/1 & = 1.0
\end{array}
$$

- We can do this with multiple trees. Simply count occurrences of CFG rules over all the trees.
- A repository of such trees labelled by a human is called a TreeBank.

Ambiguity

- Part of Speech ambiguity
  `saw` → noun
  `saw` → verb
- Structural ambiguity: Prepositional Phrases
  `I saw (the man) with the telescope`
  `I saw (the man with the telescope)`
- Structural ambiguity: Coordination
  `a program to promote safety in ((trucks) and (minivans))`
  `a program to promote ((safety in trucks) and (minivans))`
  `((a program to promote safety in trucks) and (minivans))`

Ambiguity ← attachment choice in alternative parses



Parsing as a machine learning problem

- S = a sentence
  T = a parse tree
  A statistical parsing model defines $P(T \mid S)$

Parsing as a machine learning problem

- S = a sentence
  T = a parse tree
  A statistical parsing model defines $P(T \mid S)$
- Find best parse: $\arg\max_T P(T \mid S)$

Parsing as a machine learning problem

- S = a sentence
  T = a parse tree
  A statistical parsing model defines $P(T \mid S)$
- Find best parse: $\arg\max_{T} P(T \mid S)$
- $P(T \mid S) = \frac{P(T,S)}{P(S)} = P(T,S)$

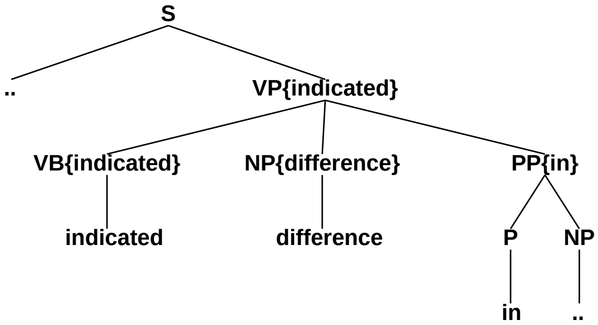
Parsing as a machine learning problem

- S = a sentence
  T = a parse tree
  A statistical parsing model defines $P(T \mid S)$
- Find best parse: $\arg\max_T P(T \mid S)$
- $P(T \mid S) = \frac{P(T,S)}{P(S)} = P(T,S)$
- Best parse: $\arg\max_T P(T,S)$

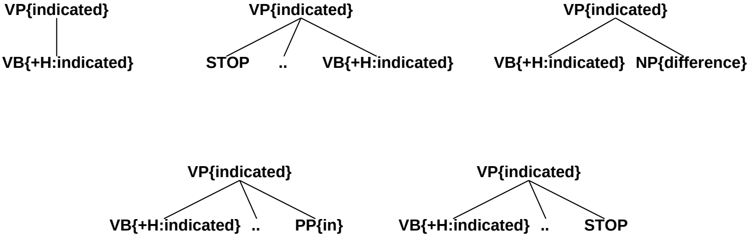
Parsing as a machine learning problem

- $S =$ a sentence
  $T =$ a parse tree
  A statistical parsing model defines $P(T \mid S)$
- Find best parse: $\underset{T}{\arg\max}\ P(T \mid S)$
- $P(T \mid S) = \frac{P(T,S)}{P(S)} = P(T,S)$
- Best parse: $\underset{T}{\arg\max}\ P(T,S)$
- e.g. for PCFGs: $P(T,S) = \prod_{i=1\ldots n} P(\mathrm{RHS}_i \mid \mathrm{LHS}_i)$

Adding Lexical Information to PCFG



Adding Lexical Information to PCFG (Collins 99, Charniak 00)



$$P_h(\mathtt{VB} \mid \mathtt{VP}, \mathtt{indicated}) \times P_l(\mathtt{STOP} \mid \mathtt{VP}, \mathtt{VB}, \mathtt{indicated}) \times$$
$$P_r(\mathtt{NP(difference)} \mid \mathtt{VP}, \mathtt{VB}, \mathtt{indicated}) \times$$
$$P_r(\mathtt{PP(in)} \mid \mathtt{VP}, \mathtt{VB}, \mathtt{indicated}) \times$$
$$P_r(\mathtt{STOP} \mid \mathtt{VP}, \mathtt{VB}, \mathtt{indicated})$$

Evaluation of Parsing

- Consider a candidate parse to be evaluated against the truth (or gold-standard parse):

```
candidate: (S (A (P this) (Q is)) (A (R a) (T test)))
gold:      (S (A (P this)) (B (Q is) (A (R a) (T test))))
```

- In order to evaluate this, we list all the constituents

Candidate	Gold
(0,4,S)	(0,4,S)
(0,2,A)	(0,1,A)
(2,4,A)	(1,4,B)
	(2,4,A)

- Skip spans of length 1 which would be equivalent to part of speech tagging accuracy.
- Precision is defined as $\frac{\#\mathit{correct}}{\#\mathit{proposed}} = \frac{2}{3}$ and recall as $\frac{\#\mathit{correct}}{\#\mathit{in\ gold}} = \frac{2}{4}$.
- Another measure: crossing brackets,

```
candidate: [ an [incredibly expensive] coat ] (1 CB)
gold:      [ an [incredibly [expensive coat]]
```

Evaluation of Parsing

$$\text{Bracketing recall } R = \frac{\text{num of correct constituents}}{\text{num of constituents in the goldfile}}$$

$$\text{Bracketing precision } P = \frac{\text{num of correct constituents}}{\text{num of constituents in the parsed file}}$$

$$\text{Complete match} = \text{\% of sents where recall \& precision are both 100\%}$$

$$\text{Average crossing} = \frac{\text{num of constituents crossing a goldfile constituent}}{\text{num of sents}}$$

$$\text{No crossing} = \text{\% of sents which have 0 crossing brackets}$$

$$\text{2 or less crossing} = \text{\% of sents which have} \leq \text{2 crossing brackets}$$

Statistical Parsing Results

$$\text{F1-score} = 2\frac{\mathit{precision} \cdot \mathit{recall}}{\mathit{precision} + \mathit{recall}}$$

System	$\leq 100\mathit{wds}$ F1-score
Shift-Reduce (Magerman, 1995)	84.14
PCFG with Lexical Features (Collins, 1999)	88.19
PCFG with Lexical Features (Charniak, 1999)	89.54
n-best Re-ranking (Collins, 2000)	89.74
Unlexicalized Berkeley parser (Petrov et al, 2007)	90.10
n-best Re-ranking (Charniak and Johnson, 2005)	91.02
Tree-insertion grammars (Carreras, Collins, Koo, 2008)	91.10
Ensemble *n*-best Re-ranking (Johnson and Ural, 2010)	91.49
Forest Re-ranking (Huang, 2010)	91.70
Unlabeled Data with Self-Training (McCloskey et al, 2006)	92.10

Practical Issues: Beam Thresholding and Priors

- Probability of nonterminal X spanning $j \dots k$: $N[X, j, k]$
- Beam Thresholding compares $N[X, j, k]$ with every other Y where $N[Y, j, k]$
- But what should be compared?
- Just the *inside probability*: $P(X \stackrel{*}{\Rightarrow} t_j \dots t_k)$? written as $\beta(X, j, k)$
- Perhaps $\beta(\texttt{FRAG}, 0, 3) > \beta(\texttt{NP}, 0, 3)$, but NPs are much more likely than FRAGs in general

Practical Issues: Beam Thresholding and Priors

- The correct estimate is the *outside probability*:

$$P(S \stackrel{*}{\Rightarrow} t_1 \dots t_{j-1}\ X\ t_{k+1} \dots t_n)$$

  written as $\alpha(X, j, k)$
- Unfortunately, you can only compute $\alpha(X, j, k)$ efficiently after you finish parsing and reach $(S, 0, n)$

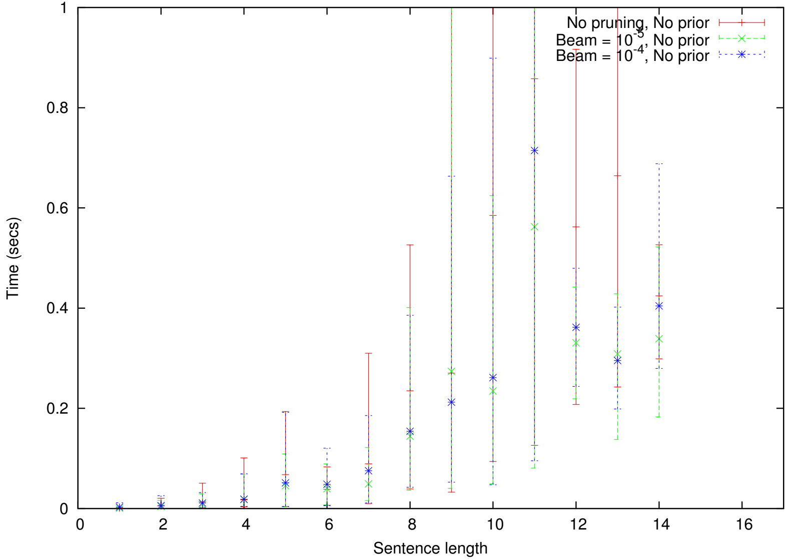
Practical Issues: Beam Thresholding and Priors

- To make things easier we multiply the prior probability $P(X)$ with the inside probability
- In beam Thresholding we compare every new insertion of X for span j, k as follows:
  Compare $P(X) \cdot \beta(X, j, k)$ with the most probable Y $P(Y) \cdot \beta(Y, j, k)$
- Assume Y is the most probable entry in j, k, then we compare

$$\text{beam} \cdot P(Y) \cdot \beta(Y, j, k) \tag{1}$$
$$P(X) \cdot \beta(X, j, k) \tag{2}$$

- If $(2) < (1)$ then we prune X for this span j, k
- beam is set to a small value, say 0.001 or even 0.01.
- As the beam value increases, the parser speed increases (since more entries are pruned).
- A simpler (but not as effective) alternative to using the beam is to keep only the top K entries for each span j, k

Experiments with Beam Thresholding



Experiments with Beam Thresholding