CMPT-413

Computational Linguistics

Anoop Sarkar

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

Probabilistic CFG (PCFG)

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

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

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

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))))
```

$$
\begin{aligned}
P(\mathit{tree}_2) &= P(S \rightarrow NP\ VP) \times P(NP \rightarrow Calvin) \times P(VP \rightarrow VP\ PP) \times \\
&\quad P(VP \rightarrow V\ NP) \times P(V \rightarrow imagined) \times P(NP \rightarrow monsters) \times \\
&\quad P(PP \rightarrow P\ NP) \times P(P \rightarrow in) \times P(NP \rightarrow school) \\
&= 1 \times 0.25 \times 0.1 \times 0.9 \times 1 \times 0.25 \times 1 \times 1 \times 0.25 = .00140625
\end{aligned}
$$

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 $P(A \rightarrow \alpha) = P(\alpha \mid A) = \frac{f(A,\alpha)}{f(A)}$
- $P(T \mid I) = \prod_i P(RHS_i \mid LHS_i)$

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

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

- 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)`

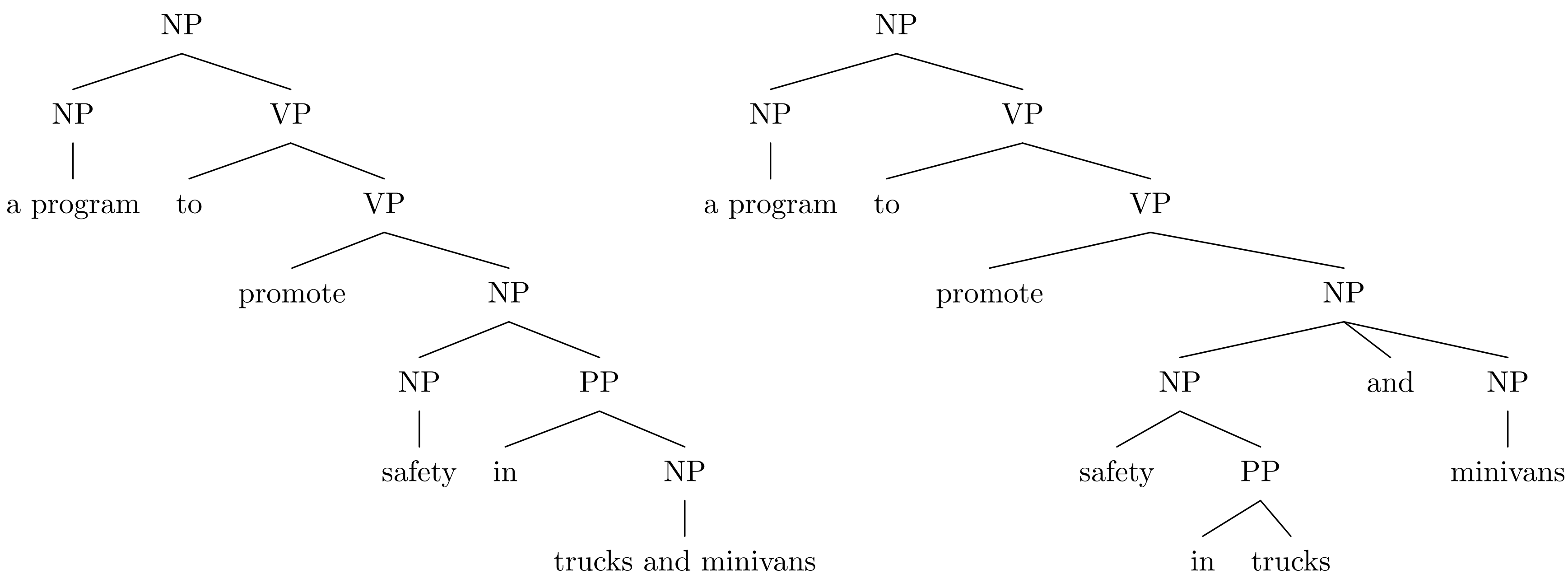
- Structural ambiguity: Coordination

  `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 $P(T \mid S)$

- 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: $\arg\max_{T} P(T, S)$

- e.g. for PCFGs: $P(T, S) = \prod_{i=1...n} P(\mathrm{RHS}_i \mid \mathrm{LHS}_i)$

Prepositional Phrases

- noun attach: *I bought the shirt with pockets*
- verb attach: *I washed the shirt with soap*
- As in the case of other attachment decisions in parsing: it depends on the meaning of the entire sentence – the so-called AI complete problem
- First we give a precise characterization of the problem and then we try to solve it using statistical associations between words

Structure Based Ambiguity Resolution

- Right association: a constituent (NP or PP) tends to attach to another constituent immediately to its right (Kimball 1973)

- Minimal attachment: a constituent tends to attach to an existing non-terminal using the fewest additional syntactic nodes (Frazier 1978)

- These two principles make opposite predictions for prepositional phrase attachment:
  e.g. in I $[_{VP}$ saw $[_{NP}$ the man ... $[_{PP}$ with the telescope],
  RA predicts that the PP attaches to the NP,
  and MA predicts VP attachment

Structure Based Ambiguity Resolution

- Garden-paths look structural:
  The emergency crews hate most is domestic violence

- Neither MA or RA account for more than 55% of the cases in real text

- Psycholinguistic experiments using eyetracking show that humans resolve ambiguities as soon as possible in the left to right sequence using the words to disambiguate

- Garden-paths are caused by a combination of lexical and structural effects:
  The flowers delivered for the patient arrived

Ambiguity Resolution: Prepositional Phrases in English

- Learning Prepositional Phrase Attachment: Annotated Data

v	n1	p	n2	Attachment
join	board	as	director	V
is	chairman	of	N.V.	N
using	crocidolite	in	filters	V
bring	attention	to	problem	V
is	asbestos	in	products	N
making	paper	for	filters	N
including	three	with	cancer	N
⋮	⋮	⋮	⋮	⋮

Prepositional Phrase Attachment

Method	Accuracy
Always noun attachment	59.0
Most likely for each preposition	72.2
Average Human (4 head words only)	88.2
Average Human (whole sentence)	93.2

Back-off Smoothing

1. If $f(v, n1, p, n2) > 0$ and $\hat{p} \neq 0.5$

$$\hat{p}(1 \mid v, n1, p, n2) = \frac{f(1, v, n1, p, n2)}{f(v, n1, p, n2)}$$

2. Else if $f(v, n1, p) + f(v, p, n2) + f(n1, p, n2) > 0$
   and $\hat{p} \neq 0.5$

$$\hat{p}(1 \mid v, n1, p, n2) = \frac{f(1, v, n1, p) + f(1, v, p, n2) + f(1, n1, p, n2)}{f(v, n1, p) + f(v, p, n2) + f(n1, p, n2)}$$

3. Else if $f(v,p) + f(n1,p) + f(p,n2) > 0$

$$\hat{p}(1 \mid v, n1, p, n2) = \frac{f(1,v,p) + f(1,n1,p) + f(1,p,n2)}{f(v,p) + f(n1,p) + f(p,n2)}$$

4. Else if $f(p) > 0$

$$\hat{p}(1 \mid v, n1, p, n2) = \frac{f(1,p)}{f(p)}$$

5. Else $\hat{p}(1 \mid v, n1, p, n2) = 1.0$

Prepositional Phrase Attachment: (Collins and Brooks 1995)

- **Results**: 84.5% accuracy
  with the use of some limited word classes for dates, numbers, etc.

- Using complex word classes taken from `WordNet` (which we shall be looking at later in this course) increases accuracy to 88%
  (Stetina and Nagao 1998)

- We can improve on parsing performance with Probabilistic CFGs by using the insights taken from PP attachment . . .

- And generalizing to other kinds of attachment problems, like coordination or deciding which constituent is an argument of a verb

Three other studies

- **Brill and Resnik, 1994**:
  use transformation based learning for PP attachment
  80.8% with words; with Wordnet classes: 81.8%
- **Toutanova, Manning, and Ng, 2004**:
  use sophisticated smoothing model for PP attachment
  86.18% with words & stems; with word classes: 87.54%
- **Merlo, Crocker and Berthouzoz, 1997**:
  test on multiple PPs, generalize disambiguation of 1 PP to 2-3 PPs
  14 structures possible for 3PPs assuming a single verb: all 14 are attested in the Treebank
  same model as CB95; but generalized to dealing with upto 3PPs
  1PP: 84.3%   2PP: 69.6%   3PP: 43.6%
  Note that this is still not the real problem faced in parsing natural language