

CMPT-825

Natural Language Processing

Anoop Sarkar

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

Probabilistic CFG (PCFG)

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

$$P_G(I) = \sum_T P_G(T \mid I)$$

$$P_G(imagined\ monsters\ in\ school) = ?$$

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

- How many different rhs α exist for $A \rightarrow \alpha$

Parsing PCFGs: CYK algorithm + the Viterbi algorithm

$\{ p: S \rightarrow S S, 1 - p: S \rightarrow a \}$

a	a	a	a
$1 - p: S_{0,1} \rightarrow a$	$1 - p: S_{1,2} \rightarrow a$	$1 - p: S_{2,3} \rightarrow a$	$1 - p: S_{3,4} \rightarrow a$
$S_{0,1} \times S_{1,2}$ $= S_{0,2} \rightarrow S S$ $p(1 - p)^2$	$S_{1,2} \times S_{2,3}$ $= S_{1,3} \rightarrow S S$ $p(1 - p)^2$	$S_{2,3} \times S_{3,4}$ $= S_{2,4} \rightarrow S S$ $p(1 - p)^2$	
$S_{0,1} + S_{1,3}$ OR $S_{0,2} + S_{2,3}$ $= S_{0,3} \rightarrow S S$ $\max(p^2(1 - p)^3,$ $p^2(1 - p)^3)$	$S_{1,2} + S_{2,4}$ OR $S_{1,3} + S_{3,4}$ $= S_{1,4} \rightarrow S S$ $\max(p^2(1 - p)^3,$ $p^2(1 - p)^3)$		
<i>What goes in this cell?</i> $?? = S_{0,4}$			

<i>S</i>	\rightarrow	<i>NP VP</i>	1
<i>VP</i>	\rightarrow	<i>V NP</i>	0.9
<i>VP</i>	\rightarrow	<i>VP PP</i>	0.1
<i>PP</i>	\rightarrow	<i>P NP</i>	1
<i>NP</i>	\rightarrow	<i>NP PP</i>	0.25
<i>NP</i>	\rightarrow	<i>Calvin</i>	0.25
<i>NP</i>	\rightarrow	<i>monsters</i>	0.25
<i>NP</i>	\rightarrow	<i>school</i>	0.25
<i>V</i>	\rightarrow	<i>imagined</i>	1
<i>P</i>	\rightarrow	<i>in</i>	1

Calvin	imagined	monsters	in	school
$NP_{0,1}$ 0.25	$V_{1,2}$ 1	$NP_{2,3}$ 0.25	$P_{3,4}$ 1	$NP_{4,5}$ 0.25
	$V_{1,2} \times NP_{2,3}$ $= VP_{1,3} \rightarrow V NP$ $1 \times 0.25 \times 0.9$		$P_{3,4} \times NP_{4,5}$ $= PP_{3,5} \rightarrow P NP$ $1^2 \times 0.25$	
	$VP_{1,3} \times PP_{3,5}$ OR $V_{1,2} \times NP_{2,5}$ $= VP_{1,5}$ x	$NP_{2,3} \times PP_{3,5}$ $= NP_{2,5} \rightarrow NP PP$ 0.25^3		
$= S_{0,4}$ $x \times 0.25$				

A Key Problem in Processing Language: Ambiguity: (Church and Patil 1982; Collins 1999)

- 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)

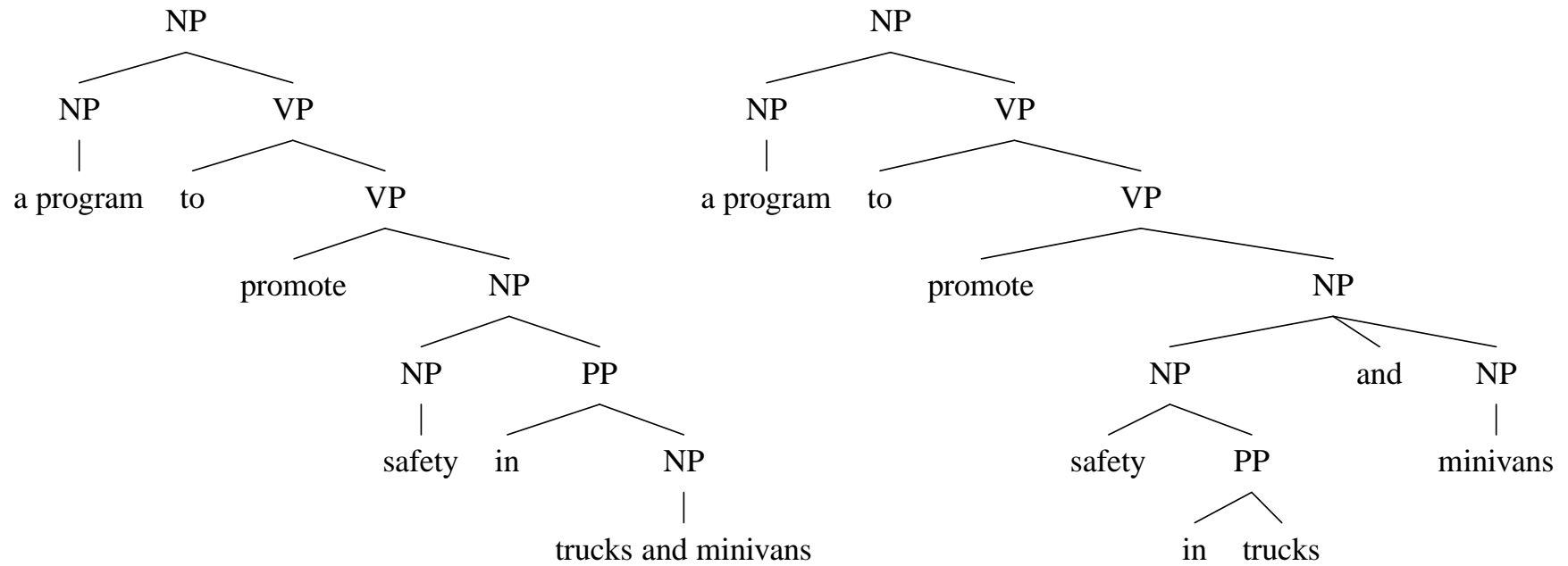
- Structural ambiguity: Coordination

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 $P(T | S)$
- Find best parse: $\arg \max_T P(T | S)$
- $P(T | S) = \frac{P(T, S)}{P(S)} = P(T, S)$
- Best parse: $\arg \max_T P(T, S)$
- e.g. for PCFGs: $P(T, S) = \prod_{i=1 \dots n} P(\text{RHS}_i | \text{LHS}_i)$

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 horse raced past the barn fell
- 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 lexical and not structural:
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

Katz 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)
- Can we improve on parsing performance using Probabilistic CFGs by using the insights detailed above

Two other studies

- **Brill and Resnik 1994:**

use transformation based learning for PP attachment

80.8% with words; with Wordnet classes: 81.8%

- **Merlo, Crocker and Berthouzoz 1997:**

test on multiple PPs, generalize the 2 PP case

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