Parsing and Discourse Parsing

Lili Mou
doublepower.mou@gmail.com

Acknowledgments

- The material basically follows Michael Collins' open course, *Natural Language Processing*, @Coursera, with extensive notes available at

  http://www.cs.columbia.edu/~mcollins/

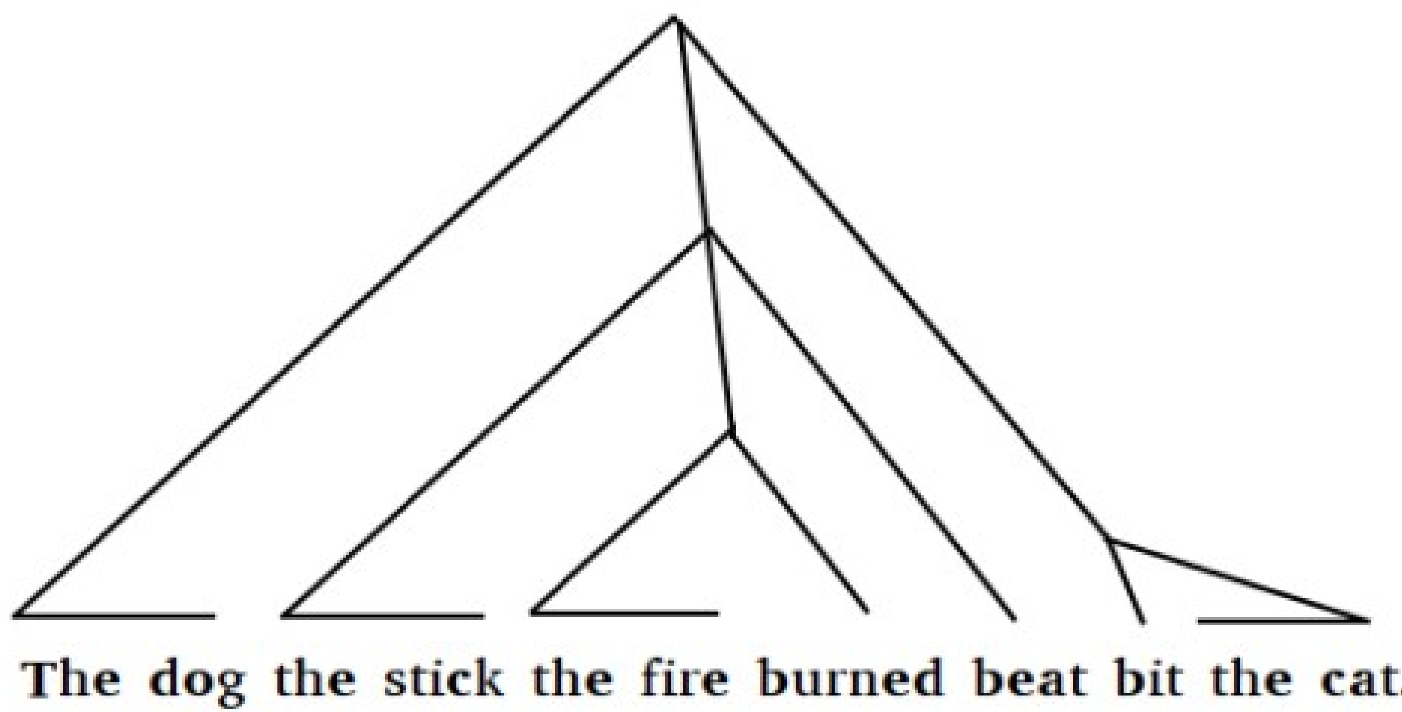
- We may also refer interested audience to Jurafsky & Martin's book, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Second Edition)*

Road Map

- The Parsing Problem
- Context-Free Grammar
- Probabilistic Context Free Grammar
- CKY Algorithm
- Lexicalized PCFG
- Discourse Parsing (Next Week)

The parsing problem

- My favorite example: Onion sentences [Pinker, 1994]
  - The dog the stick the fire burned beat bit the cat.
  - If if if it rains it pours I get depressed I should get help.
  - That that that he left is apparent is clear is obvious.



Road Map

- The Parsing Problem
- Context-Free Grammar
- Probabilistic Context-Free Grammar
- CKY Algorithm
- Lexicalized PCFG
- Discourse Parsing (Next Week)

Context Free Grammar

A context-free grammar (CFG) is a 4-tuple $G = (N, \Sigma, R, S)$ where:

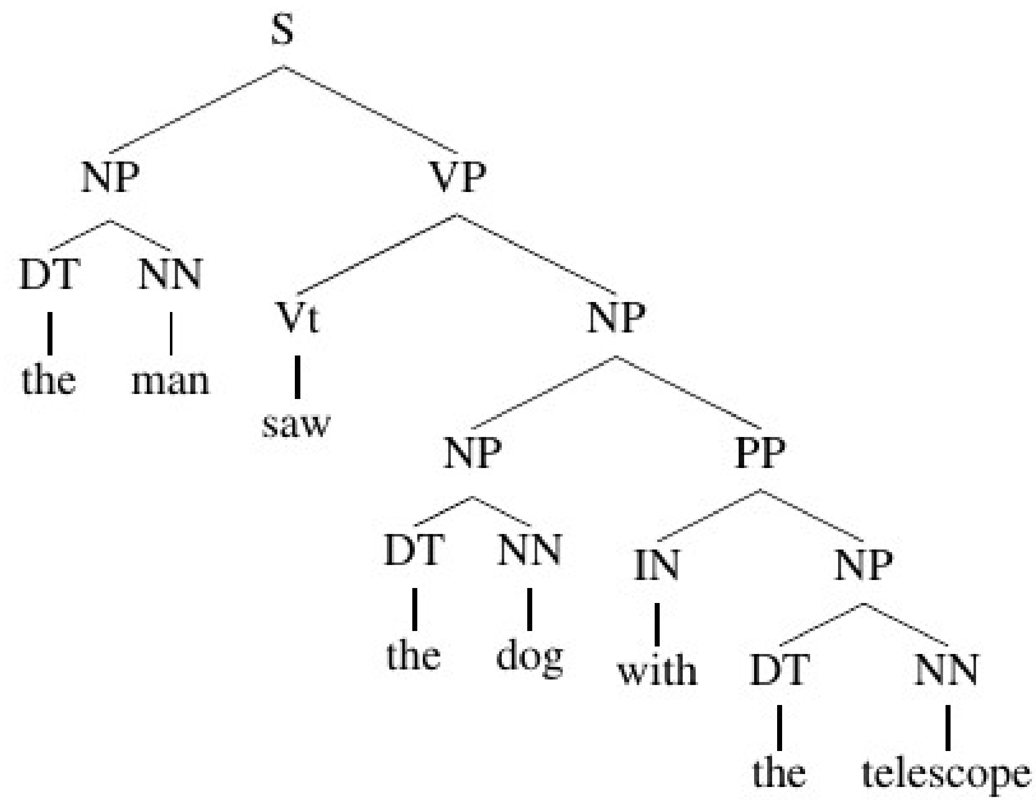
- N is a finite set of non-terminal symbols.
- Σ is a finite set of terminal symbols.
- R is a finite set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$, where $X \in N$, $n \geq 0$, and $Y_i \in (N \cup \Sigma)$ for $i = 1 \ldots n$.
- $S \in N$ is a distinguished start symbol.

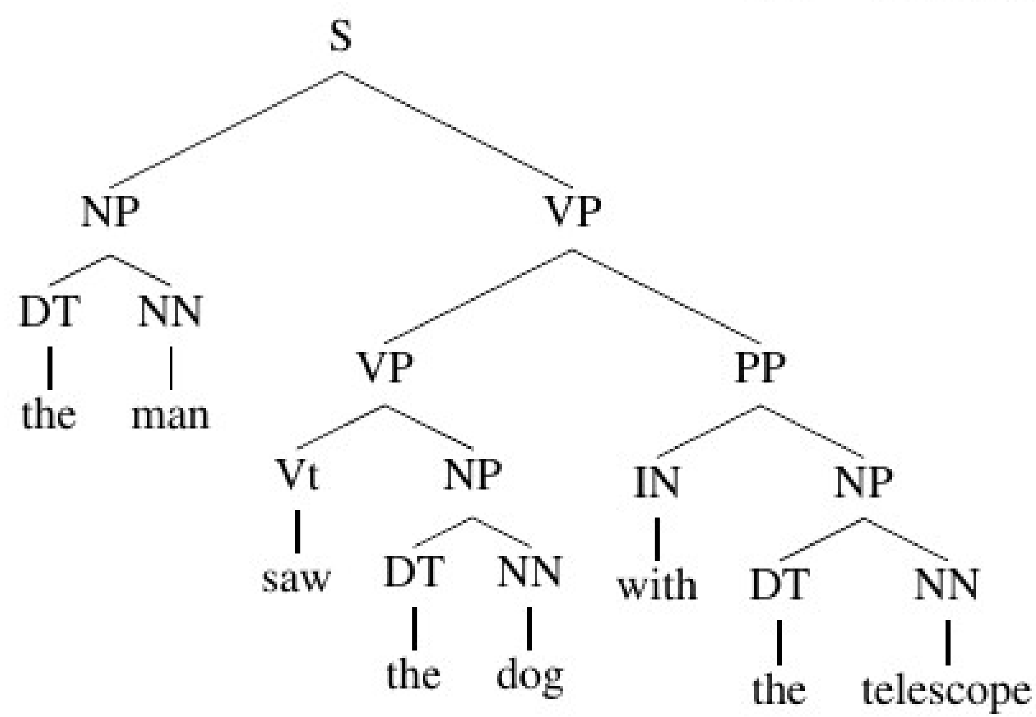
E.g.,
- S: a sentence
- N: NP, VP, PP
- ∑: a word
- R: S--> NP VP, NP--> N, N-->book

Parsing in PLP and NLP

- Parsing a program
  - The syntax of a programming language guarantees no ambiguity.
  - It usually also guarantees an efficient (greedy) parsing algorithm, e.g., LALR.
- Parsing a natural language sentence
  - Ambiguity is usually the No. 1 concern.
  - Many of possible results don't make sense (w/ low probability).

The man saw the dog with the telescope.





Road Map

- The Parsing Problem
- Context-Free Grammar
- Probabilistic Context-Free Grammar
- CKY Algorithm
- Lexicalized PCFG
- Discourse Parsing (Next Week)

Probabilistic Context-Free Grammar

Definition 1 (PCFGs) *A PCFG consists of:*

1. *A context-free grammar* $G = (N, \Sigma, S, R)$.

2. *A parameter*

$$q(\alpha \rightarrow \beta)$$

for each rule $\alpha \rightarrow \beta \in R$. *The parameter* $q(\alpha \rightarrow \beta)$ *can be interpreted as the conditional probabilty of choosing rule* $\alpha \rightarrow \beta$ *in a left-most derivation, given that the non-terminal being expanded is* α. *For any* $X \in N$, *we have the constraint*

$$\sum_{\alpha \rightarrow \beta \in R : \alpha = X} q(\alpha \rightarrow \beta) = 1$$

In addition we have $q(\alpha \rightarrow \beta) \geq 0$ *for any* $\alpha \rightarrow \beta \in R$.

- PCFG in a nutshell:

  PCFG is nothing but a CFG with probability assigned to each rule.

Example

$N = \{$S, NP, VP, PP, DT, Vi, Vt, NN, IN$\}$
S = S
$\Sigma = \{$sleeps, saw, man, woman, dog, telescope, the, with, in$\}$

$R, q =$

S	$\rightarrow$	NP	VP	1.0
VP	$\rightarrow$	Vi		0.3
VP	$\rightarrow$	Vt	NP	0.5
VP	$\rightarrow$	VP	PP	0.2
NP	$\rightarrow$	DT	NN	0.8
NP	$\rightarrow$	NP	PP	0.2
PP	$\rightarrow$	IN	NP	1.0

Vi	$\rightarrow$	sleeps	1.0
Vt	$\rightarrow$	saw	1.0
NN	$\rightarrow$	man	0.1
NN	$\rightarrow$	woman	0.1
NN	$\rightarrow$	telescope	0.3
NN	$\rightarrow$	dog	0.5
DT	$\rightarrow$	the	1.0
IN	$\rightarrow$	with	0.6
IN	$\rightarrow$	in	0.4

- saw with a telescope v.s. dog with a telescope
- VP--> VP PP and NP → NP PP may have different probabilities in general.

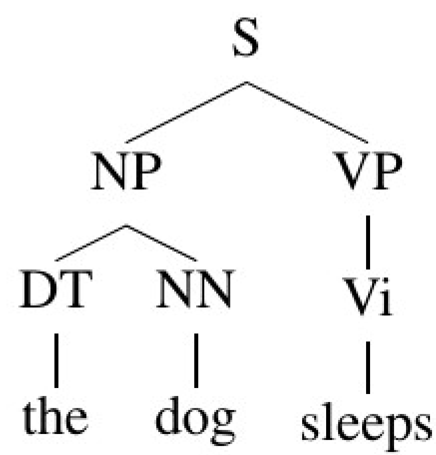
The probability of a parse tree

- Under very mild conditions

Given a parse-tree $t \in \mathcal{T}_G$ containing rules $\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \ldots, \alpha_n \rightarrow \beta_n$, the probability of t under the PCFG is

$$p(t) = \prod_{i=1}^{n} q(\alpha_i \rightarrow \beta_i)$$

- Example



then we have

$$\begin{aligned} p(t) \quad = \quad & q(\texttt{S} \rightarrow \texttt{NP VP}) \times q(\texttt{NP} \rightarrow \texttt{DT NN}) \times q(\texttt{DT} \rightarrow \texttt{the}) \times q(\texttt{NN} \rightarrow \texttt{dog}) \times \\ & q(\texttt{VP} \rightarrow \texttt{Vi}) \times q(\texttt{Vi} \rightarrow \texttt{sleeps}) \end{aligned}$$

Learning/Paramter Estimation

- Maximum likelihood estimation

Road Map

- The Parsing Problem
- Context-Free Grammar
- Probabilistic Context-Free Grammar
- CKY Algorithm
- Lexicalized PCFG
- Discourse Parsing (Next Week)

Prediction/Decoding

- The parsing problem is to find a parse tree that

$$\arg\max_{t \in \mathcal{T}_G(s)} p(t)$$

Preprocessing

Definition 2 (Chomsky Normal Form) *A context-free grammar* $G = (N, \Sigma, R, S)$ *is in Chomsky form if each rule* $\alpha \rightarrow \beta \in R$ *takes one of the two following forms:*

- $X \rightarrow Y_1 Y_2$ *where* $X \in N, Y_1 \in N, Y_2 \in N.$
- $X \rightarrow Y$ *where* $X \in N,\ Y \in \Sigma.$

- Binarize

The CKY Algorithm

- The Cocke–Younger–Kasami algorithm
- Dynamic programming
- Pi(i, j, X): The max. probability of non-terminal symbol X spanning words from i to j.

$$\pi(i, j, X) = \max_{t \in \mathcal{T}(i,j,X)} p(t)$$

- Initialization:

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{if } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

- Recursion:

$$\pi(i,j,X) = \max_{\substack{X \to YZ \in R, \\ s \in \{i \dots (j-1)\}}} \left(q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)$$

back-pointer

$$bp(i,j,X) = \arg \max_{\substack{X \to YZ \in R, \\ s \in \{i \dots (j-1)\}}} \left(q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)$$

- Termination:

$$\pi(1,n,S) = \max_{t \in \mathcal{T}(s)} p(t)$$

Road Map

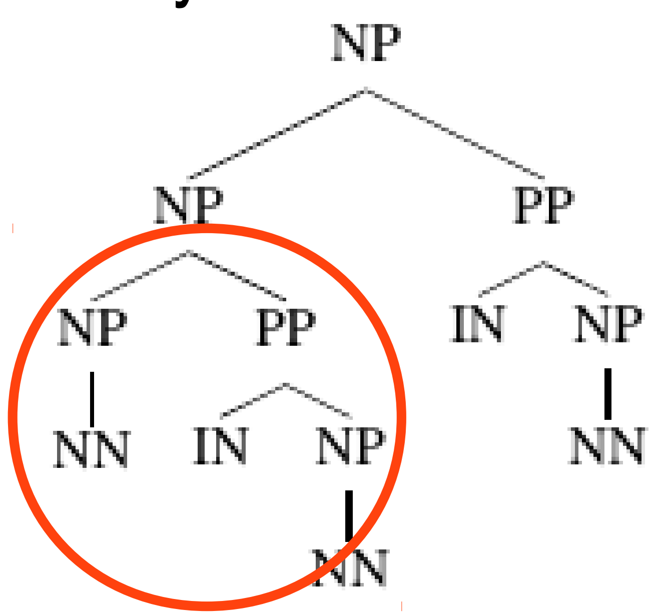
- The Parsing Problem
- Context-Free Grammar
- Probabilistic Context-Free Grammar
- CKY Algorithm
- Lexicalized PCFG
- Discourse Parsing (Next Week)

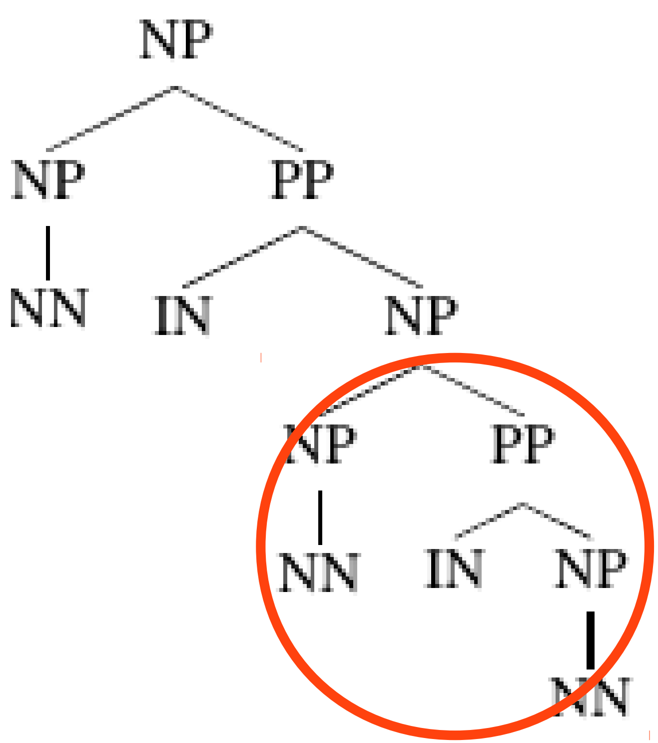
Weakness of PCFG

- Lack of Sensitivity to Lexical Information
  - The man saw the dog with a telescope
  - The man saw the girl with a telescope

  The only difference lies in N-> dog or N->girl, which is a constant given either of the above sentences. However, a girl is more likely to wield a telescope than a dog.

- Lack of Sensitivity to Structural Preferences



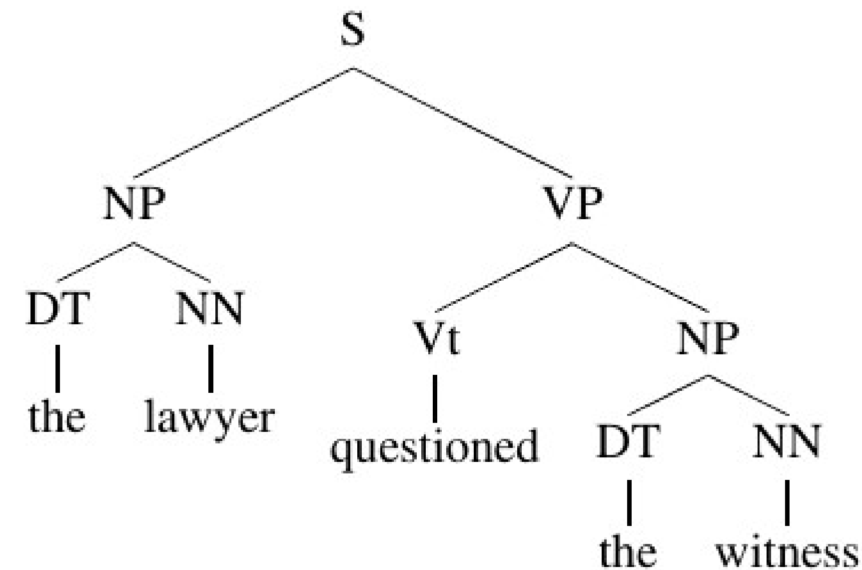


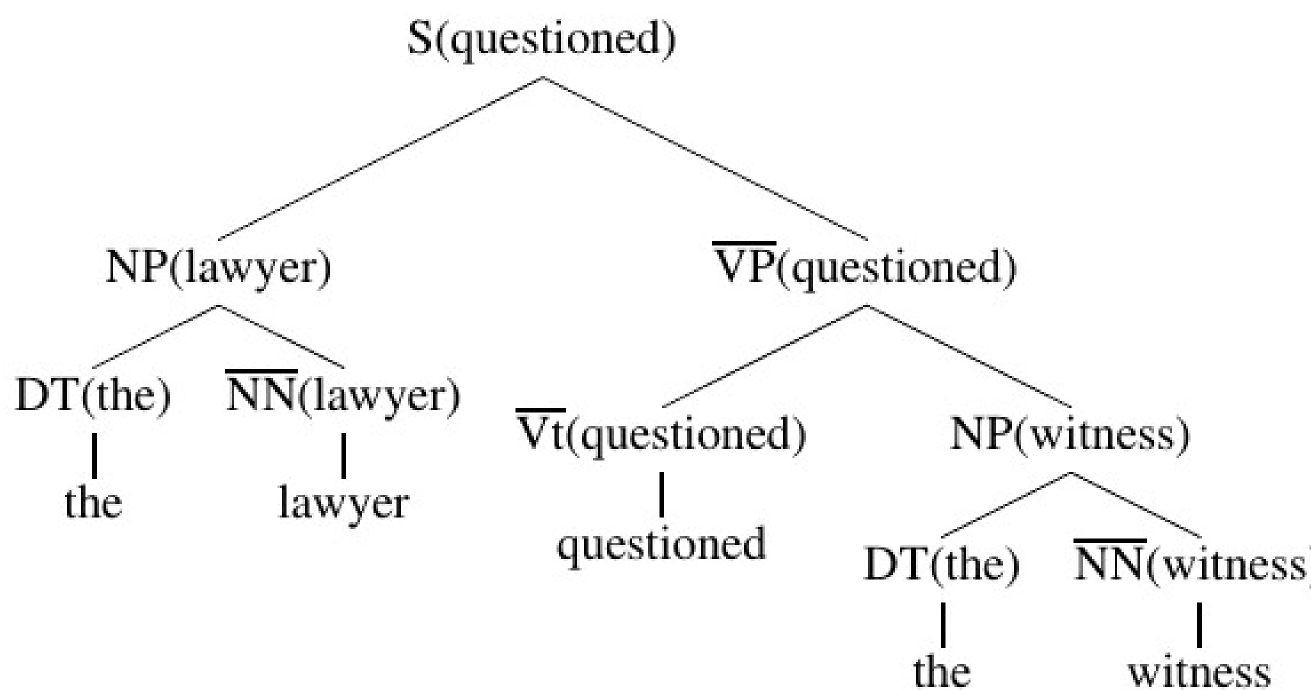
Lexicalization

- Tag each abstract constituent with a word from its child nodes (by heuristics)
- S--> NP $\overline{VP}$,

  where $\overline{}$ refers

  to the headword





- Side product:

  dependency relations

Rules to Tag the Headword

If the rule contains NN, NNS, or NNP:
    Choose the rightmost NN, NNS, or NNP

Else If the rule contains an NP: Choose the leftmost NP

Else If the rule contains a JJ: Choose the rightmost JJ

Else If the rule contains a CD: Choose the rightmost CD

Else Choose the rightmost child

Figure 6: Example of a set of rules that identifies the head of any rule whose left-hand-side is an NP.

Learning/Parameter Estimation

- E.g., $q(S(\text{examined}) \rightarrow NP(\text{lawyer})\ VP(\text{examined}))$

- Maximum likelihood estimation
- Smoothing

Prediction/Decoding

- Dynamic programming
- Pi(i, j, h, X): The max. probability of constituent X with head h, spanning over word i to j.
- Initialization:

$$\pi(i,i,i,X) = \begin{cases} q(X(x_i) \rightarrow x_i) & \text{if } X(x_i) \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

- Recursion:

For $s = h \ldots (j-1)$, for $m = (s+1) \ldots j$, for $X(x_h) \rightarrow_1 Y(x_h)Z(x_m) \in R$,

(a) $p = q(X(x_h) \rightarrow_1 Y(x_h)Z(x_m)) \times \pi(i, s, h, Y) \times \pi(s+1, j, m, Z)$

(b) If $p > \pi(i, j, h, X)$,

$$\pi(i, j, h, X) = p$$

$$bp(i, j, h, X) = \langle s, m, Y, Z \rangle$$

For $s = i \ldots (h-1)$, for $m = i \ldots s$, for $X(x_h) \rightarrow_2 Y(x_m)Z(x_h) \in R$,

(a) $p = q(X(x_h) \rightarrow_2 Y(x_m)Z(x_h)) \times \pi(i, s, m, Y) \times \pi(s+1, j, h, Z)$

(b) If $p > \pi(i, j, h, X)$,

$$\pi(i, j, h, X) = p$$

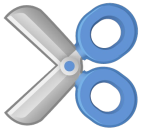
$$bp(i, j, h, X) = \langle s, m, Y, Z \rangle$$

- The headword may come from either the left child or the right child

- Termination

$$(X^*, h^*) = \arg \max_{S \in N, h \in \{1 \ldots n\}} \gamma(X, h) \times \pi(1, n, h, X)$$

Road Map

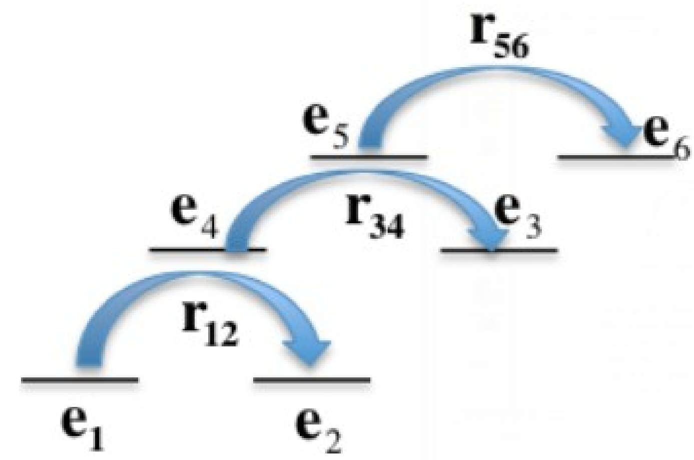
- The Parsing Problem
- Context-Free Grammar
- Probabilistic Context Free Grammar
- CKY Algorithm
- Lexicalized PCFG
- Discourse Parsing 

Recursive Deep Models for Discourse Parsing

Revisit

Jiwei Li[1], Rumeng Li[2] and Eduard Hovy[3]
[1]Computer Science Department, Stanford University, Stanford, CA 94305, USA
[2]School of EECS, Peking University, Beijing 100871, P.R. China
[3]Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
jiweil@stanford.edu alicerumeng@foxmail.com ehovy@andrew.cmu.edu

- Dataset: Rhetorical Structure Theory Discourse Treebank (RST-DT)
- 385 documents, 347 for training (5-fold), 49 for testing
- Each doc represented as a tree
  - Elementary Discourse Units (EDUs): Clauses
  - Relations: hypotactic v.s. paratactic



- EDU modeling: Standard RAE
- Discourse parsing:
- 2-step strategy
  - Binary classifier: To determine whether two adjacent text units should be merged to form a new subtree

$$t_{\text{binary}}(e_1, e_2) = 1, \quad t_{\text{binary}}(e_3, e_4) = 1,$$
$$t_{\text{binary}}(e_2, e_3) = 0, \quad t_{\text{binary}}(e_3, e_6) = 0,$$
$$t_{\text{binary}}(e_5, e_6) = 1$$

$$L^{\text{binary}}_{(e_i, e_j)} = f(G_{\text{binary}} * [h_{e_i}, h_{e_j}] + b_{\text{binary}})$$

$$p[t_{\text{binary}}(e_i, e_j) = 1] = g(U_{binary} \cdot L^{\text{binary}}_{(e_i, e_j)} + b^{*}_{\text{binary}})$$

  - Multi-class classifier: To determine which relation

Learning/Parameter Estimation

- Whether two EDUs have some relation?

$$J(\Theta_{\text{binary}}) = \sum_{(e_i,e_j)\in\{\text{binary}\}} J_{\text{binary}}(e_i, e_j) + Q_{\text{binary}} \cdot \sum_{\theta\in\Theta_{\text{binary}}} \theta^2$$

- And what relation?

$$J(\Theta_{\text{multi}}) = \sum_{(e_i,e_j)\in\{\text{multi}\}} J_{\text{multi}}(e_i, e_j) + Q_{\text{multi}} \cdot \sum_{\theta\in\Theta_{\text{multi}}} \theta^2$$

Inference/Decoding

- Choose the parse tree with max. prob.
- Dynamic programming, keeping 10 options at each time

  Pr[r, i, j]: The max. prob. that (discourse) relation r spans over EDUs i to j.

$$
\begin{aligned}
Pr[r,i,j] = &\max_{r_1,r_2,k} Pr[r_1,i,k] \cdot Pr[r_2,k,j] \\
&\times P(t_{\text{binary}}(e_{[i,k]}, e_{[k,j]}) = 1) \\
&\times P(r(e_{[i,k]}, e_{[k,j]}) = 1)
\end{aligned}
$$