Phân tích cú pháp xác suất

Lê Thanh Hương
Bộ môn Hệ thống Thông tin
Viện CNTT &TT – Trường ĐHBKHN
Email: huonglt-fit@mail.hut.edu.vn

Làm cách nào chọn cây đúng?

- Ví dụ:
  I saw a man with a telescope.
- Khi số luật tăng, khả năng nhập nhằng tăng
- Tập luật NYU: bộ PTCP Apple pie : 20,000-30,000 luật cho tiếng Anh
- Lựa chọn luật AD: V DT NN PP
  (1) VP → V NP PP
  NP → DT NN
  (2) VP → V NP
  NP → DT NN PP

Kết hợp từ (bigrams pr)

Ví dụ:
Eat ice-cream (high freq)
Eat John (low, except on Survivor)

Nhược điểm:

- P(John decided to bake a) có xác suất cao
- Xét:
  $P(w_3) = P(w_3|w_2w_1)=P(w_3|w_2)P(w_2|w_1)P(w_1)$

Giả thiết này quá mạnh: chủ ngữ có thể quyết định bổ ngữ trong câu
Clinton admires honesty

➢ sử dụng cấu trúc ngữ pháp để dừng việc lan truyền

- Xét Fred watered his mother's small garden. Từ *garden* có ảnh hưởng như thế nào?
  - Pr(*garden*|*mother's small*) thấp ⇒ mô hình trigram không tốt
  - Pr(*garden* | X là thành phần chính của bổ ngữ cho động từ *to water*) cao hơn

➢ sử dụng bigram + quan hệ ngữ pháp

Kết hợp từ (bigrams pr)

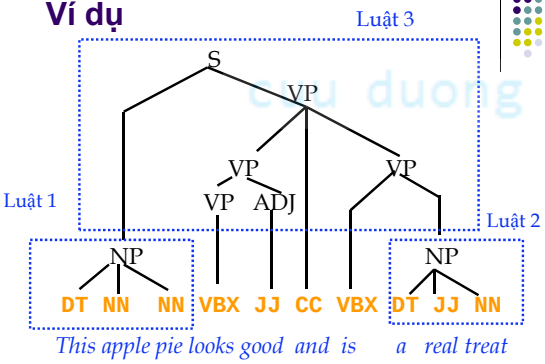
- V có một số loại bổ ngữ nhất định
  ⇒ Verb-with-obj, verb-without-obj
- Sự tương thích giữa chủ ngữ và bổ ngữ:
  John admires honesty
  Honesty admires John ???

Nhược điểm:

- Kích thước tập ngữ pháp tăng
- Các bài báo của tạp chí Wall Street Journal trong 1 năm: 47,219 câu, độ dài trung bình 23 từ, gán nhãn bằng tay: chỉ có 4.7% hay 2,232 câu có cùng cấu trúc ngữ pháp

➢ Không thể dựa trên việc tìm các cấu trúc cú pháp đúng cho cả câu. Phải xây dựng tập các mẫu ngữ pháp nhỏ

Ví dụ



Luật

1. NP→DT NN NN
2. NP→DT JJ NN
3. S→NP VBX JJ CC VBX NP

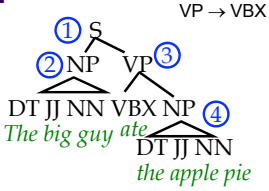
- Nhóm (NNS, NN) thành NX; (NNP, NNPs)=NPX; (VBP, VBZ, VBD)=VBX;
- Chọn các luật theo tần suất của nó

Tính xác suất

$$\Pr(X \rightarrow Y) = \frac{\text{X}}{\text{Y}} \Rightarrow \frac{\text{NP}}{\text{DT JJ NN}} = \frac{\text{NP}}{\text{NP}} = \frac{1470}{9711} = 0.1532$$

Tính Pr

S → NP VP; 0.35
NP → DT JJ NN; 0.1532
VP → VBX NP; 0.302



Luật áp dụng	*Chuỗi Pr*
1 S →NP VP	0.35
2 NP → DT JJ NN	0.1532 x 0.35 = 0.0536
3 VP → VBX NP	0.302 x 0.0536= 0.0162
4 NP → DT JJ NN	0.1532 x 0.0162=0.0025

Pr = 0.0025

Văn phạm phi ngữ cảnh xác suất

- 1 văn phạm phi ngữ cảnh xác suất (Probabilistic Context Free Grammar) gồm các phần thông thường của CFG
- Tập ký hiệu kết thúc $\{w^k\}$, k = 1, . . . ,V
- Tập ký hiệu không kết thúc $\{N^i\}$, i = 1, . . . ,n
- Ký hiệu khởi đầu N^1
- Tập luật $\{N^i \rightarrow \zeta^j\}$, ζ^j là chuỗi các ký hiệu kết thúc và không kết thúc
- Tập các xác suất của 1 luật là:
  $\forall i \sum_j P(N^i \rightarrow \zeta^j) = 1$
- Xác suất của 1 cây cú pháp:
  $P(T) = \Pi_{i=1..n}\, p(r(i))$

Các giả thiết

- ***Độc lập vị trí:*** Xác suất 1 cây con không phụ thuộc vào vị trí của các từ của cây con đó ở trong câu

  $\forall k, P(N_{jk(k+c)} \rightarrow \zeta)$ là giống nhau

- ***Độc lập ngữ cảnh:*** Xác suất 1 cây con không phụ thuộc vào các từ ngoài cây con đó

  $P(N_{jkl} \rightarrow \zeta |$ các từ ngoài khoảng k đến l) = $P(N_{jkl} \rightarrow \zeta)$

- ***Độc lập tổ tiên:*** Xác suất 1 cây con không phụ thuộc vào các nút ngoài cay con đó

  $P(N_{jkl} \rightarrow \zeta |$ các nút ngoài cây con N_{jkl}) = $P(N_{jkl} \rightarrow \zeta)$

Các thuật toán

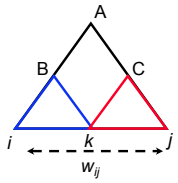
- CKY
- Beam search
- Agenda/chart-based search
- …

CKY kết hợp xác suất

- Cấu trúc dữ liệu:
  - Mảng lập trình động $\pi[i,j,a]$ lưu xác suất lớn nhất của ký hiệu không kết thúc a triển khai thành chuỗi i…*j*.
  - Backptrs lưu liên kết đến các thành phần trên cây
- Ra: Xác suất lớn nhất của cây

Tính Pr dựa trên suy diễn

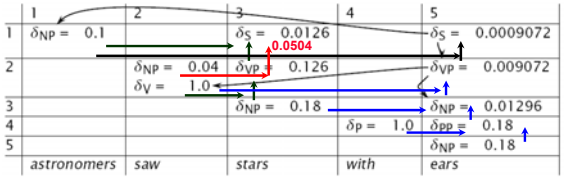
- Trường hợp cơ bản: chỉ có 1 từ đầu vào
  Pr(tree) = pr($A \rightarrow w_i$)
- Trường hợp đệ qui: Đầu vào là xâu các từ
  $A \overset{*}{\Rightarrow} w_{ij}$ if $\exists k$: $A \rightarrow BC$, $B \overset{*}{\Rightarrow} w_{ik}$,$C \overset{*}{\Rightarrow} w_{kj}$,$i \le k \le j$.
  $p[i,j] = \max(p(A \rightarrow BC) \times p[i,k] \times p[k,j])$.




```
function CYK(words,grammar) returns best_parse

Create and clear p[num_words,num_words,num_nonterminals]

# base case
for i = 1 to num_words
  for A = 1 to num_nonterminals
    if A → w_i is in grammar then
      π[i,i,A] = P(A → w_i)

# recursive case
for j = 2 to num_words
  for i = 1 to num_words-j+1
    for k = 1 to j-1
      for A = 1 to num_nonterminals
      for B = 1 to num_nonterminals
      for C = 1 to num_nonterminals
        prob = π[i,k,B] × p[i+k,j-k,C] × P(A→BC)
        if (prob > π[i,j,A]) then
          π[i,j,A] = prob
          B[i,j,A] = {k,A,B}
```


Tính xác suất Viterbi (thuật toán CKY)

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1



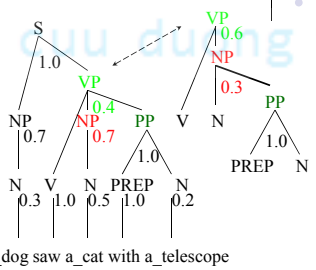
Ví dụ

• S → NP VP	0.80	• Det → the	0.50
• NP → Det N	0.30	• Det → a	0.40
• VP → V NP	0.20	• N → meal	0.01
• V → includes	0.05	• N → flight	0.02

Dùng thuật toán CYK phân tích câu vào:
"The flight includes a meal"

Tính Pr

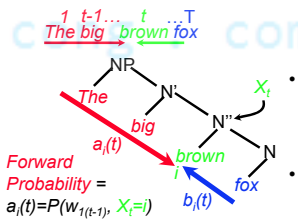
1.	S → NP VP	1.0
2.	VP → V NP PP	0.4
3.	VP → V NP	0.6
4.	NP → N	0.7
5.	NP → N PP	0.3
6.	PP → PREP N	1.0
7.	N → a_dog	0.3
8.	N → a_cat	0.5
9.	N → a_telescop	0.2
10.	V → saw	1.0
11.	PREP → with	1.0



a_dog saw a_cat with a_telescope

$P_l = 1 \times .7 \times .4 \times .3 \times .7 \times 1 \times .5 \times 1 \times 1 \times .2 = .00588$
$P_r = 1 \times .7 \times .6 \times .3 \times .3 \times 1 \times .5 \times 1 \times 1 \times .2 = .00378$
➢ P_l is chosen

Xác suất Forward và Backward

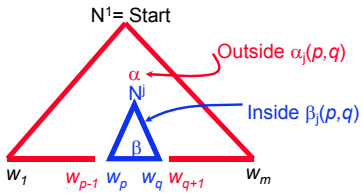


Forward Probability = $a_i(t)=P(w_{1(t-1)}, X_t=i)$

Backward Probability = $b_i(t)=P(w_{tT} \mid X_t=i)$

- *Forward*= xác suất các phần tử trên và bao gồm 1 nút cụ thể nào đó
- *Backward*= xác suất các phần tử dưới 1 nút cụ thể nào đó

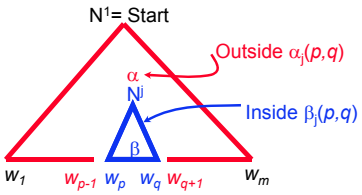
Xác suất trong và ngoài



- N_{pq} = ký hiệu không kết thúc N^j trải từ vị trí p đến q trong xâu
- α_j = xác suất ngoài (outside)
- β_j = xác suất trong (inside)
- N^j *phủ* các từ $w_p \ldots w_q$, *nếu* $N^j \Rightarrow^* w_p \ldots w_q$

19

Xác suất trong và ngoài



$\alpha_j(p,q) = P(w_{1(p-1)}, N_{pq}^j, w_{(q+1)m} | G)$

$\beta_j(p,q) = P(w_{pq} | N_{pq}^j, G)$

$\alpha_j(p,q)\ \beta_j(p,q) = P(N^1 \Rightarrow^* w_{1m}, N^j \Rightarrow^* w_{pq} | G)$

$= P(N^1 \Rightarrow^* w_{1m} | G) \bullet P(N^j \Rightarrow^* w_{pq} | N^1 \Rightarrow^* w_{1m}, G)$

20

Tính xác suất của xâu

- Sử dụng thuật toán ***Inside***, 1 thuật toán lập trình động dựa trên xác suất inside

$P(w_{1m}|G) = P(N^1 \Rightarrow^* w_{1m}|G) = P(w_{1m}|N_{1m}^1, G) = \beta_1(1,m)$

- Trường hợp cơ bản:

  $\beta_j(k,k) = P(w_k|N_{kk}^j, G) = P(N^j \rightarrow w_k|G)$

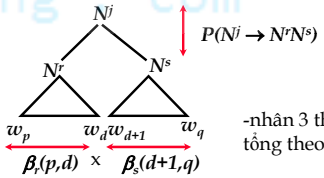
- Suy diễn:

  $\beta_j(p,q) = \Sigma_{r,s}\Sigma_{d\in(p,q-1)}\ P(N^j \rightarrow N^r N^s)\ \beta_r(p,d)\ \beta_s(d+1,q)$

21

Suy diễn

Tính $\beta_j(p,q)$ với p < q – tính trên tất cả các điểm j – thực hiện từ dưới lên

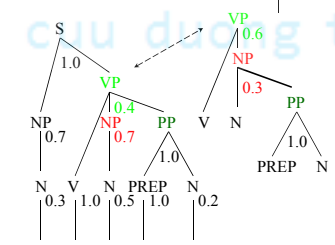


-nhân 3 thành phần, tính tổng theo *j, r,s.*

22

Ví dụ

1.	S → NP VP	1.0
2.	VP → V NP PP	0.4
3.	VP → V NP	0.6
4.	NP → N	0.7
5.	NP → N PP	0.3
6.	PP → PREP N	1.0
7.	N → a_dog	0.3
8.	N → a_cat	0.5
9.	N → a_telescope	0.2
10.	V → saw	1.0
11.	PREP → with	1.0



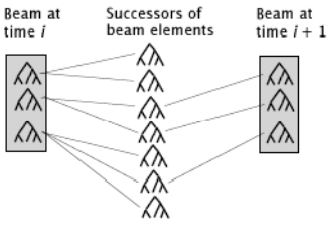
P(a_dog saw a_cat with a_telescope) =

1×.7×.4×.3×.7×1×.5×1×1×.2 + ... ×.6... ×.3... = .00588 + .00378 = .00966

23

Tìm kiếm kiểu chùm

- Tìm kiếm trong không gian trạng thái
- Mỗi trạng thái là một cây cú pháp con với 1 xác suất nhất định
  - Tại mỗi thời điểm, chỉ giữ các thành phần có điểm cao nhất



24

Làm giàu PCFG

- PCFG đơn giản hoạt động không tốt do các giả thiết độc lập
- Giải quyết: Đưa thêm thông tin
  - Phụ thuộc cấu trúc
    - Việc triển khai 1 nút phụ thuộc vào vị trí của nó trên cây (độc lập với nội dung về từ vựng của nó)
    - Ví dụ: bổ sung thông tin cho 1 nút bằng cách lưu giữ thông tin về cha của nó: ^{S}NP khác với ^{VP}NP

25

Làm giàu PCFG

- PCFG từ vựng hóa : PLCFG (Probabilistic Lexicalized CFG, Collins 1997; Charniak 1997)
- Gán từ vựng với các nút của luật
- Cấu trúc Head
  - Mỗi phần tử của parsed tree được gắn liền với một *lexical head*
  - Để xác định *head* của một nút trong ta phải xác định trong các nút con, nút nào là *head* (xác định *head* trong vế phải của một luật).

26

Làm giàu PLCFG

VP(dumped) → VBD(dumped) NP(sacks) PP(into) $3*10^{-10}$
VP(dumped) → VBD(dumped) NP(cats) PP(into) $8*10^{-11}$

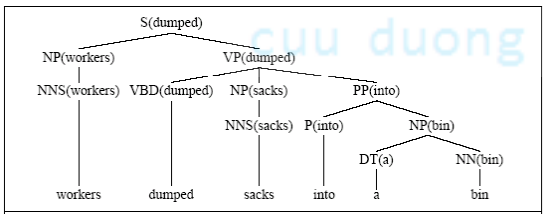


Figure 12.12 A lexicalized tree from Collins (1999).

27

Tại sao dùng PLCFG

- Tính ngoại lệ (exception) của ngôn ngữ
- Sự phân loại theo cú pháp hiện tại chưa thể hiện hết đặc tính hoạt động của từng từ vựng.
- Từ vựng hóa luật CFG giúp bộ phân tích cú pháp thực hiện chính xác hơn

Hạn chế của PLCFG

VP -> VBD NP PP
VP(*dumped*) -> VBD(*dumped*) NP(*sacks*) PP(*into*)

- Không có một corpus đủ lớn!
  - Thể hiện hết các trường hợp cú pháp, hết các trường hợp đối với từng từ.

Penn Treebank

- Penn Treebank: tập ngữ liệu có chú giải ngữ pháp, có 1 triệu từ, là nguồn ngữ liệu quan trọng
- Tính thưa:
  - có 965,000 mẫu, nhưng chỉ có 66 mẫu WHADJP, trong đó chỉ có 6 mẫu không là *how much* hoặc *how many*
- Phần lớn các phép xử lý thông minh phụ thuộc vào các thống kê mối quan hệ từ vựng giữa 2 từ liền nhau:

30

A Penn Treebank tree

```
( (S (NP-SBJ The move)
     (VP followed
         (NP (NP a round)
             (PP of
                 (NP (NP similar increases)
                     (PP by
                         (NP other lenders))
                     (PP against
                         (NP Arizona real estate loans)))))
         ,
         (S-ADV (NP-SBJ *)
                (VP reflecting
                    (NP (NP a continuing decline)
                        (PP-LOC in
                                (NP that market))))))
     .))
```

Đánh giá độ chính xác của PTCP

- Độ chính xác của parser được đo qua việc tính xem có bao nhiêu thành phần ngữ pháp trong cây giống với cây chuẩn, gọi là gold-standard reference parses.
- Độ chính xác (Precision) =

$$\% \frac{\text{trường hợp hệ gán đúng}}{\text{tổng số trường hợp hệ gán}}$$

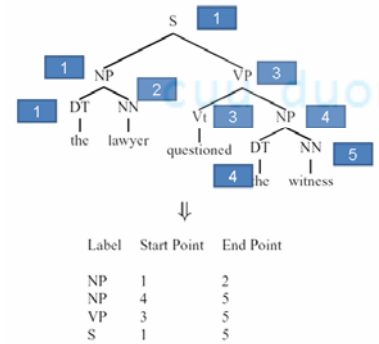
(%THợp hệ tính đúng).

- Độ phủ (Recall) =

$$\% \frac{\text{số trường hợp hệ gán đúng}}{\text{tổng số trường hợp đúng}}$$

(%THợp hệ tính đúng so với con người).

Biểu diễn cây theo các thành phần ngữ pháp



Label	Start Point	End Point
NP	1	2
NP	4	5
VP	3	5
S	1	5

Đánh giá

Precision and Recall

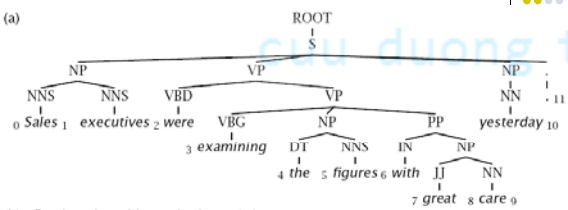
Label	Start Point	End Point
NP	1	2
NP	4	5
NP	4	8
PP	6	8
NP	7	8
VP	3	8
S	1	8

Label	Start Point	End Point
NP	1	2
NP	4	5
PP	6	8
NP	7	8
VP	3	8
S	1	8

- G = number of constituents in gold standard = 7
- P = number in parse output = 6
- C = number correct = 6

$$\text{Recall} = 100\% \times \frac{C}{G} = 100\% \times \frac{6}{7} \qquad \text{Precision} = 100\% \times \frac{C}{P} = 100\% \times \frac{6}{6}$$

Ví dụ 2



(b) Brackets in gold standard tree (a.):
S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), **NP-(4:6),** PP-(6-9), NP-(7,9), *NP-(9:10)

(c) Brackets in candidate parse:
S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:10), **NP-(4:6),** PP-(6-10), NP-(7,10)

(d)

Precision:	3/8 = 37.5%	Crossing Brackets:	0
Recall:	3/8 = 37.5%	Crossing Accuracy:	100%
Labeled Precision:	3/8 = 37.5%	Tagging Accuracy:	10/11 = 90.9%
Labeled Recall:	3/8 = 37.5%		

Độ chính xác của các hệ thống PTCP

Method	Recall	Precision
PCFGs (Charniak 97)	70.6%	74.8%
Conditional Models – Decision Trees (Magerman 95)	84.0%	84.3%
Lexical Dependencies (Collins 96)	85.3%	85.7%
Conditional Models – Logistic (Ratnaparkhi 97)	86.3%	87.5%
Generative Lexicalized Model (Charniak 97)	86.7%	86.6%
Model 1 (no subcategorization)	87.5%	87.7%
Model 2 (subcategorization)	88.1%	88.3%