



人工知能特論II

二宮 崇

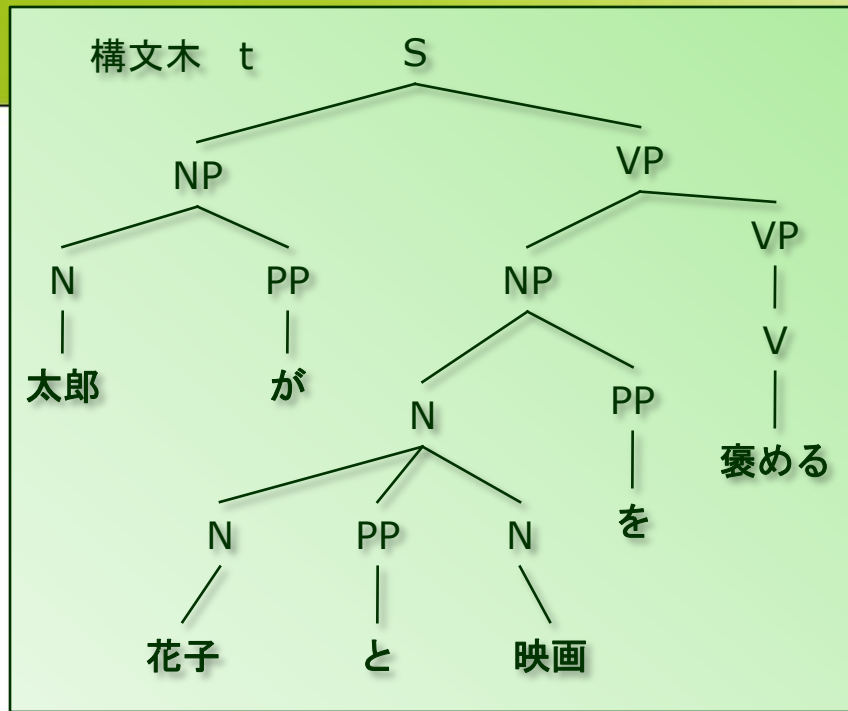
今日の講義の予定

- PCFG (Probabilistic Context Free Grammar, 確率付文脈自由文法)
- マルコフ文法
- 論文
 - Michael Collins (1997) Three generative, lexicalised models for statistical parsing, In proc. of ACL-EACL, p.16 – 23
 - Michael Collins (1999) Head-Driven Statistical Models for Natural Language Parsing, Ph.D thesis, University of Pennsylvania
 - Michael Collins (2003) Head-Driven Statistical Models for Natural Language Parsing, Computational Linguistics, 29(4), 589--637.
 - Daniel M. Bikel (2004) Intricacies of Collins' Parsing Model, Computational Linguistics, 30, 479 – 511



構文木tの確率を計算

CFG G	パラメータ	値
S → NP VP	$\theta_{S \rightarrow NP VP}$	1.0
NP → N PP	$\theta_{NP \rightarrow N PP}$	1.0
N → N PP N	$\theta_{N \rightarrow N PP N}$	0.1
VP → NP VP	$\theta_{VP \rightarrow NP VP}$	0.3
VP → V	$\theta_{VP \rightarrow V}$	0.7
PP → が	$\theta_{PP \rightarrow が}$	0.5
PP → を	$\theta_{PP \rightarrow を}$	0.3
PP → と	$\theta_{PP \rightarrow と}$	0.2
N → 太郎	$\theta_{N \rightarrow 太郎}$	0.3
N → 花子	$\theta_{N \rightarrow 花子}$	0.2
N → 映画	$\theta_{N \rightarrow 映画}$	0.4
V → 褒める	$\theta_{V \rightarrow 褒める}$	0.3
V → 見る	$\theta_{V \rightarrow 見る}$	0.7



$$\begin{aligned}
 p(t) &= \theta_{S \rightarrow NP VP} \times \theta_{NP \rightarrow N PP} \times \theta_{N \rightarrow 太郎} \times \\
 &\quad \theta_{PP \rightarrow が} \times \theta_{VP \rightarrow NP VP} \times \theta_{NP \rightarrow N PP} \times \\
 &\quad \theta_{N \rightarrow N PP N} \times \theta_{N \rightarrow 花子} \times \theta_{PP \rightarrow と} \times \\
 &\quad \theta_{N \rightarrow 映画} \times \theta_{PP \rightarrow を} \times \theta_{VP \rightarrow V} \times \theta_{V \rightarrow 褒める} \\
 &= 1.0 \times 1.0 \times 0.3 \times 0.5 \times 0.3 \times 1.0 \times \\
 &\quad 0.1 \times 0.2 \times 0.2 \times 0.4 \times 0.3 \times 0.7 \times 0.3 \\
 &= 0.000004536
 \end{aligned}$$

最大確率の木を選ぶ (構文解析)

- ある文 s に対し、CFG $\langle V_N, V_T, P, \sigma \rangle$ を用いて s を導出できる全ての構文木集合を $T(s)$ としたとき、

$$\tilde{t} = \arg \max_{t \in T(s)} p(t)$$



パラメータ推定 (学習)

- 単純な数え上げ
 - 教師付学習のもっとも単純な実現
 - 正解構文木の集合(ツリーバンクと呼ばれる)があれば、その中で使われているCFG書換規則の頻度をカウントすれば各々のパラメータを簡単に計算できる
- 最尤推定
 - 教師無学習の最も有名な実現
 - 正解構文木がない場合や書換規則パラメータ以外のパラメータが存在する場合に使われる



パラメータは条件付き確率

- $A \rightarrow \alpha$ という形のCFGルールに対するパラメータ $\theta_{A \rightarrow \alpha}$
- パラメータの関数と明示するほうがわかりやすいけど、その確率は親が生成された時の条件確率
 - $\theta_{A \rightarrow \alpha} = p(\alpha | A)$
- 今後は、パラメータの形ではなくて、条件付き確率の形式で説明していきます



PCFG概説

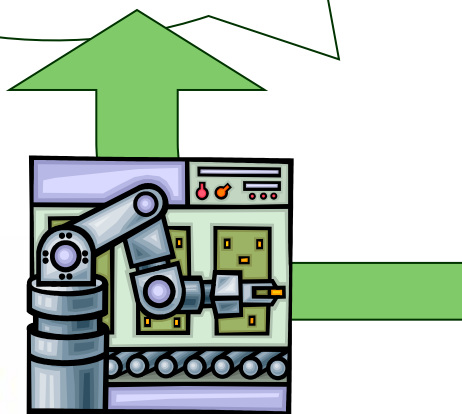


ツリーバンクを使った構文解析

S → NP VP
NP → DET N
NP → N
...



文法規則、辞書



コンピュータ

- 人間が文法を直接定義するのは困難
- 構文木の实例（ツリーバンク）に基づく定量的評価が必要
- 文法はツリーバンクから導出

(ツリーバンク文法)



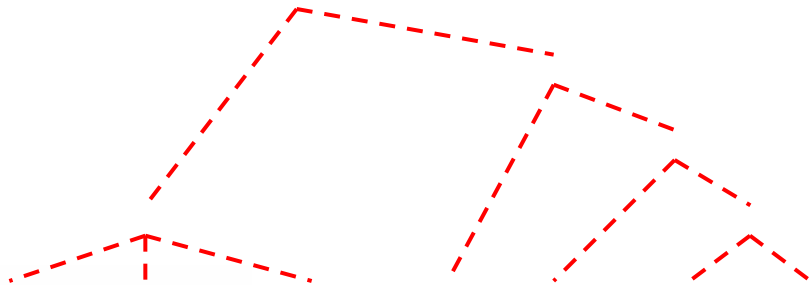
ツリーバンク

検証・開発



ツリーバンク

- 実世界の文に対して人手で構文木を付与する
- 明示的な文法を仮定しない
- 構造は開発者の言語直感とガイドラインに依存
 - ガイドラインはあるが、文法で定義されるような「何が正解か」の客観的基準は存在しない



A record date has n't been set.



有名なツリーバンク

- 構文木や係り受け木を人手で付与したコーパス（ツリーバンク）の登場
 - Penn Treebank [Marcus et al. 1993]
 - SUSANNE [Sampson 1995]
 - TIGER Treebank [Brants et al. 2002]
 - Prague Dependency Treebank [Hajic 1998]
 - Verbmobil [Hinrichs et al. 2000]
 - EDRコーパス [EDR 1995]
 - 京都大学テキストコーパス [黒橋ら 1997]
 - 日本語話し言葉コーパス [前川ら 2000]



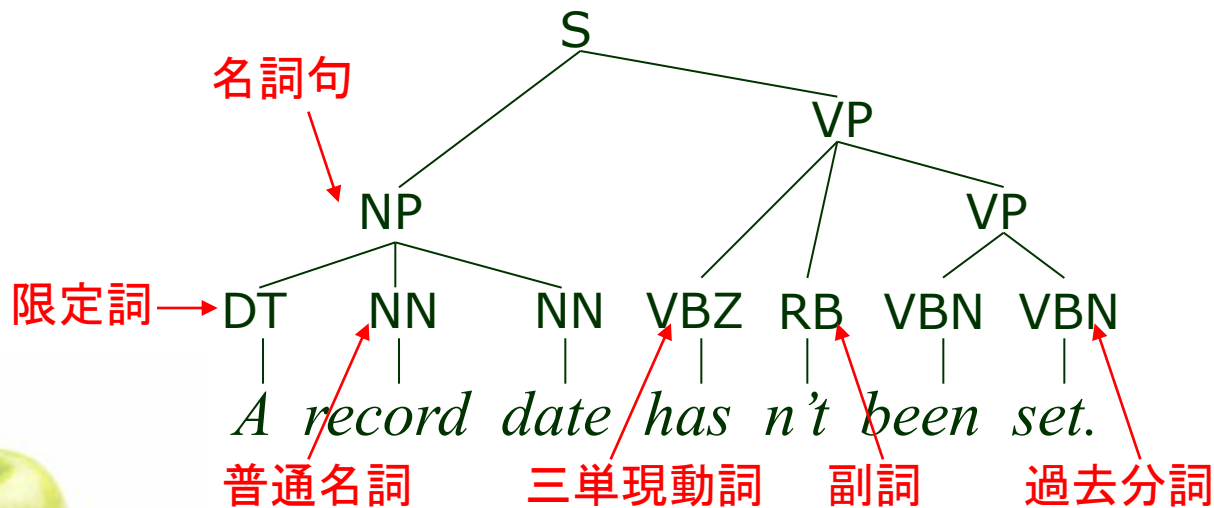
Penn Treebank (1/2)

- 構文木が付与された最初の大規模英語ツリーバンク [Marcus et al. 1993]
- 様々な分野の英語テキストを収録
 - Wall Street Journal (新聞) 約5万文、100万語
 - ATIS (航空券予約の会話)
 - Brown (様々な分野のテキスト)
 - Switchboard (電話の自由発話)



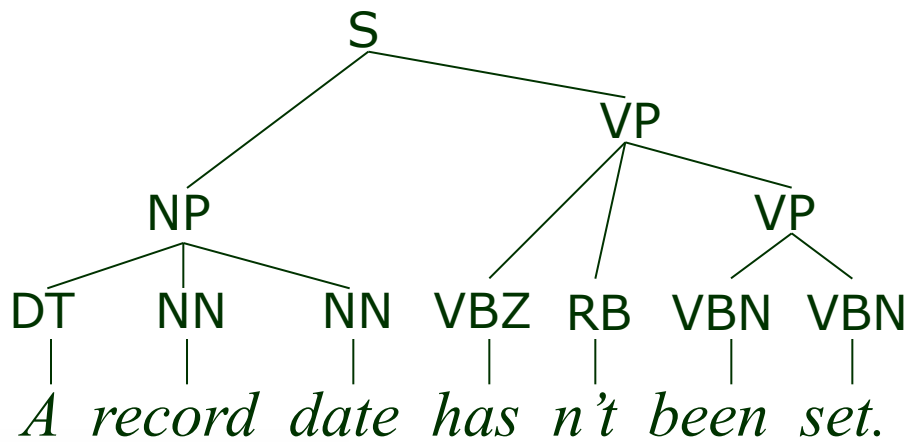
Penn Treebank (2/2)

- 品詞：NN（普通名詞），VBZ（三単現動詞） ...
- 構文木：NP（名詞句），VP（動詞句） ...
- Function tag, null element: 述語項構造を計算するための付加情報（詳細省略）

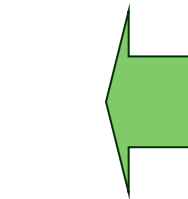
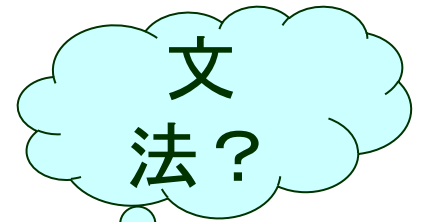


ツリーバンクから文法を抽出する

- ツリーバンクの背後にある文法を自動抽出
 - 潜在的な規則性を自動獲得できるはず



文法抽出

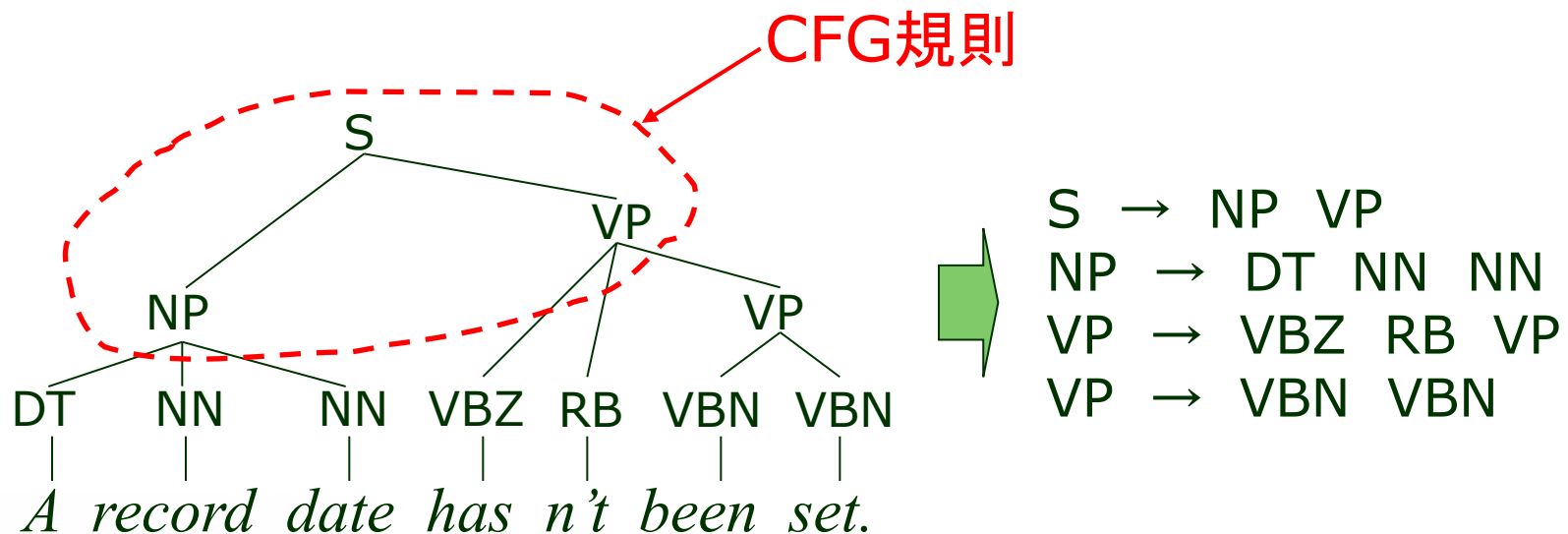


ツリーバンク
開発



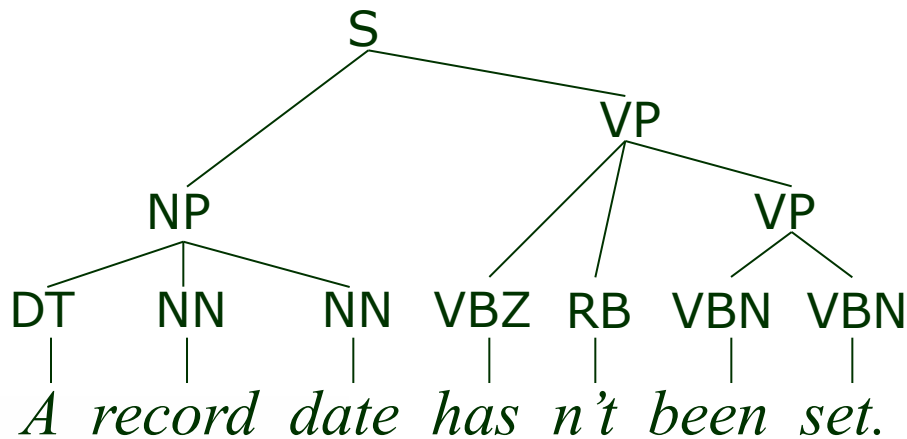
確率CFGの自動抽出(1/2)

- ツリーバンクの各分岐をCFG規則だと仮定して抽出する [Charniak 1996; 1997] c.f. [Sekine1995]



確率CFGの自動抽出(2/2)

- ツリーバンクでの出現頻度から確率値を推定
- 確率値最大の木を探索することで、構文解析の曖昧性解消ができる

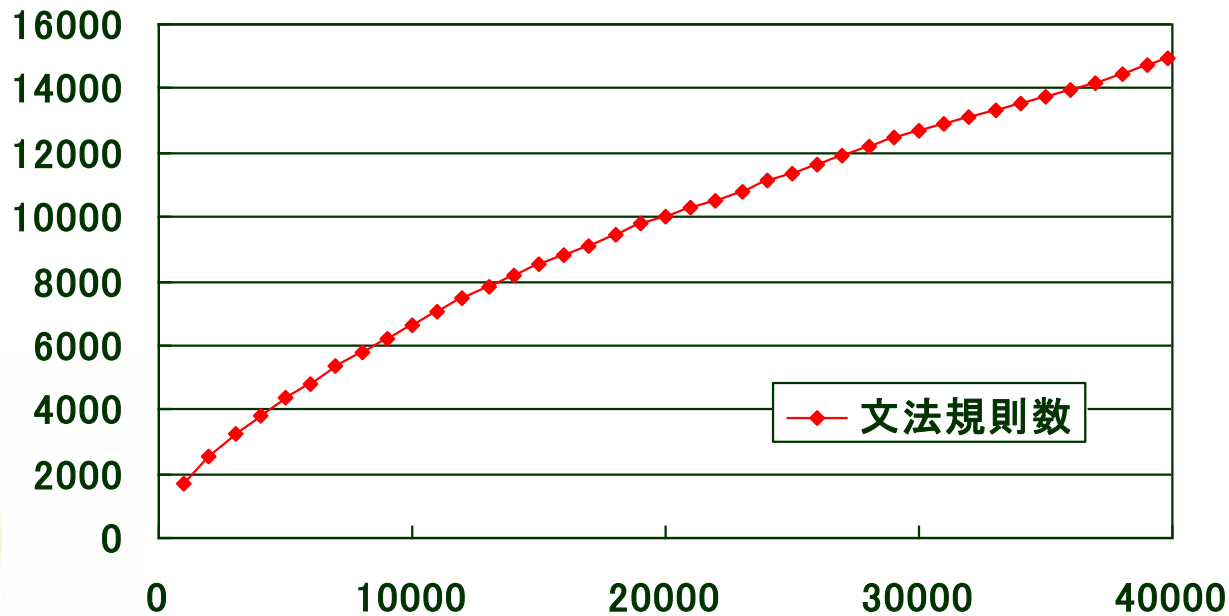


S	→	NP	VP	0.5	
NP	→	DT	NN	NN	0.03
VP	→	VBZ	RB	VP	0.02
VP	→	VBN	VBN	0.1	



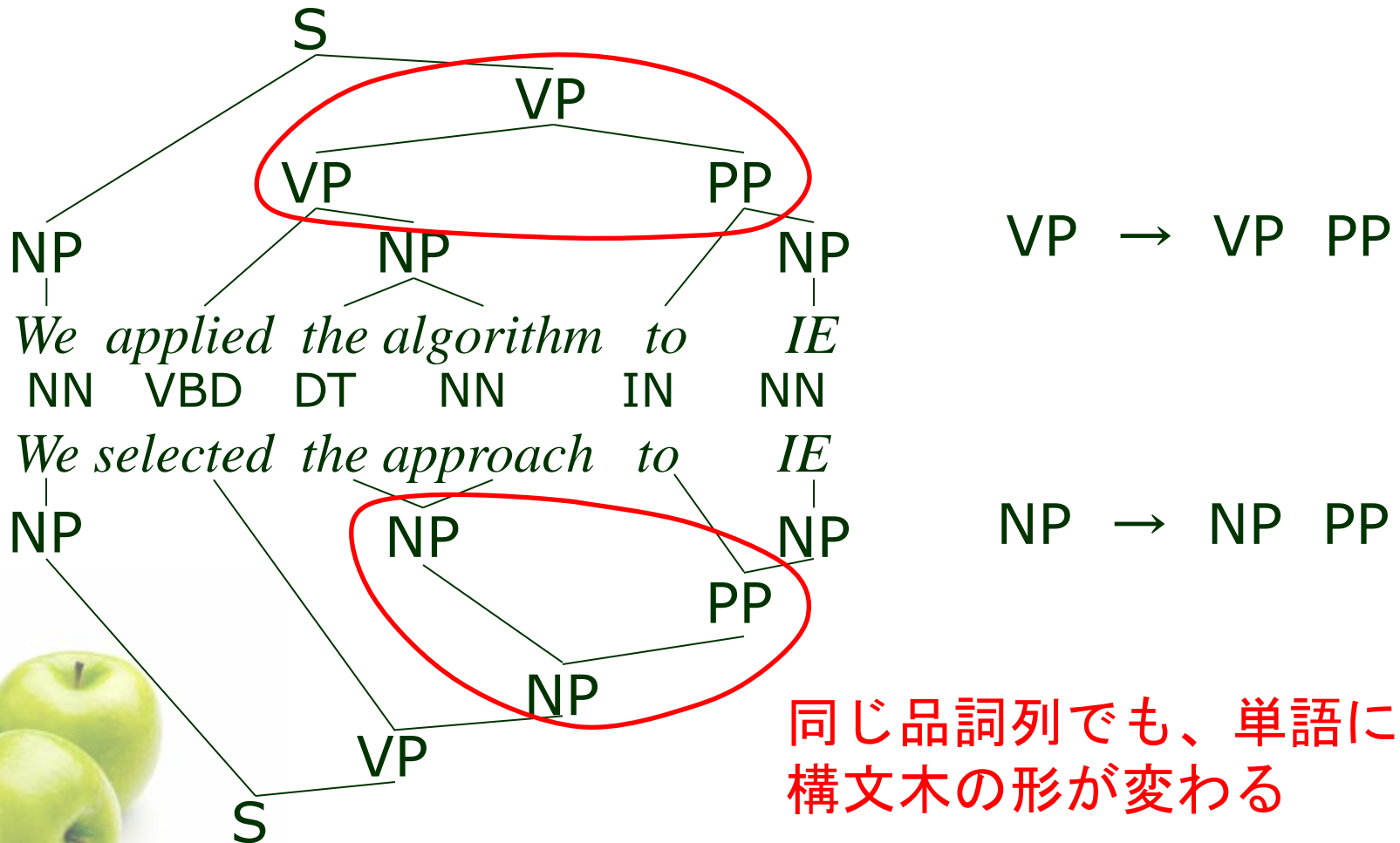
問題点（1）：文法が大きい

- 40,000文から約15,000のCFG規則
 - CFG規則数が収束しない [Carpenter et al. 1997]
- 抽象化・一般化しきれていない



問題点（２）：精度が低い

- Charniak [1996]: 80%



ツリーバンク文法の改良

(1) 文法が大きい

- CFG規則の自動圧縮 [Krotoy et al. 1998; 1999]
- CFG規則の確率モデル化 [Magerman 1995; Collins 1997; Charniak 2000]

(2) 精度が低い

- 非終端記号の細分化 [Magerman 1995; Collins 1996; 1997; Johnson 1998; Charniak 2000]



CFG規則の確率モデル化

- Markov Grammar: CFG規則を確率的に生成する [Collins 1997; Charniak 2000]

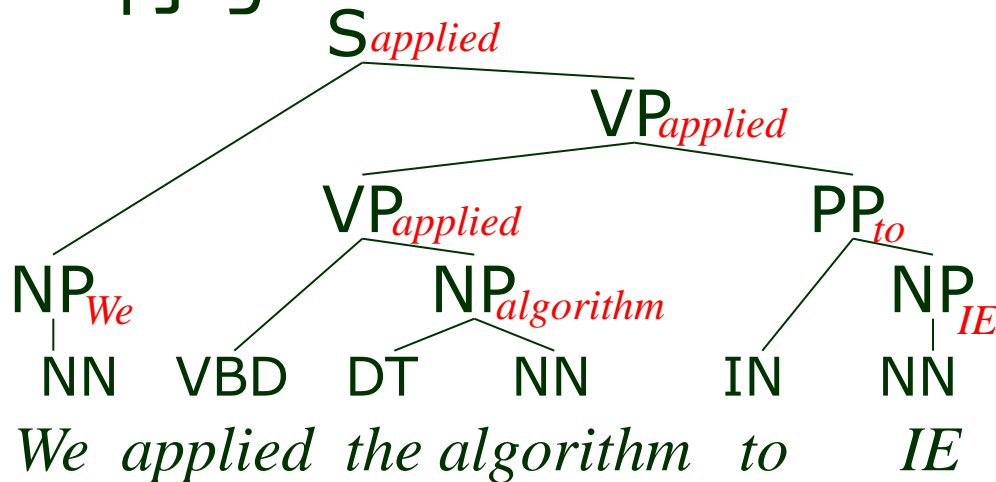
$$\begin{aligned} & p(\text{NP} \rightarrow \text{DT NN NN} \mid \text{NP}) \\ &= p(\text{NN} \mid \text{NP}) p(\text{NN} \mid \text{NN, NP}) p(\text{DT} \mid \text{NN, NN, NP}) \end{aligned}$$

- 原理的には、全てのCFG規則をもつ PCFG
- Penn Treebank から抽出したそのままの PCFG より高精度を達成する



非終端記号の細分化(1/2)

- 語彙化: Head percolation table (Magerman 1995) を用いて、非終端記号に head word を付与



Head percolation table

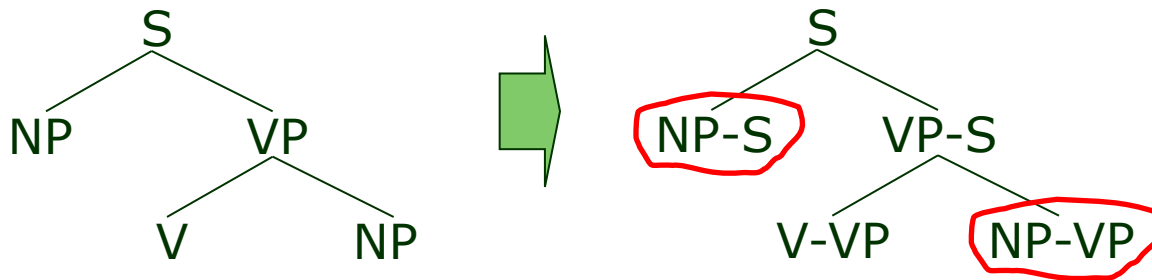
親の記号	主辞になる子の記号
S	VP, ...
VP	VP, VBD, VBZ, ...
NP	NN, ...
PP	IN, ...

Charniak [1996]: 80% vs. Magerman [1995]: 86%

- (参考) 語彙化の意味 [Gildea 2001; Bikel 2004]

非終端記号の細分化(2/2)

- 非終端記号だけでは構造を決める情報が少ない
- (例) 親の非終端記号で細分化 [Johnson 1998]



- 主語のNPと目的語のNPが区別できる
 - 主語は代名詞が出やすい
 - 目的語は長くなりやすい
- その他、様々な周辺情報で細分化 [Charniak 2000; Klein et al. 2003]

マルコフ文法



マルコフ文法

- CFG規則を確率的に生成する [Collins 1997; Charniak 2000]

$$\begin{aligned} & p(\text{NP} \rightarrow \text{DT NN NN} \mid \text{NP}) \\ &= p(\text{NN} \mid \text{NP}) p(\text{NN} \mid \text{NN, NP}) p(\text{DT} \mid \text{NN, NN, NP}) \end{aligned}$$

- 原理的には、全てのCFG規則をもつ PCFG
- Penn Treebank から抽出したそのままの PCFG より高精度を達成する
- State-of-the-artの性能のパースーの基本的な仕組

何故マルコフ文法を講義で扱うのか？

- 現在の（おおよそ）最高性能のパースー
(Charniak&Johnson2005)の基礎
 - Charniak(2000)のパースーの出力をエントロピー最大化法を用いてreranking
- Charniakパースーもマルコフ文法の一つ

モデル	精度(LF)
collins1999	88.19%
charniak2000	89.55%
charniak&johnson2005	91.02%



マルコフ過程: シャノンゲーム

- Nana eats an ????
 - 次に何が来るのか予測

バイグラム

$$P(\text{apple} | \text{an}) = \frac{C(\text{an apple})}{C(\text{an})}$$

$$P(\text{apple} | \text{eats an}) = \frac{C(\text{eats an apple})}{C(\text{eats an})}$$

トライグラム



マルコフ過程: 確率モデル

- 条件付き確率は

$$p(B | A) = \frac{p(A, B)}{p(A)}$$

- だから、

$$p(A, B) = p(A)p(B | A)$$

- つまり、

$$p(A_1) = p(A_1)$$

$$p(A_1, A_2) = p(A_1)p(A_2 | A_1)$$

$$p(A_1, A_2, A_3) = p(A_1, A_2)p(A_3 | A_1, A_2) = p(A_1)p(A_2 | A_1)p(A_3 | A_1, A_2)$$

$$p(A_1, \dots, A_n) = p(A_1)p(A_2 | A_1)p(A_3 | A_1, A_2) \cdots p(A_n | A_1, \dots, A_{n-1})$$



マルコフ過程: 確率モデル

- 単語列の確率モデル

$$p(w_1, \dots, w_n) = p(w_1)p(w_2 | w_1)p(w_3 | w_1, w_2)p(w_4 | w_1, w_2, w_3) \cdots p(w_n | w_1, \dots, w_{n-1})$$

- 単語列のNグラムモデル(N-1次のマルコフ過程)

- 直前のN-1個の単語列の影響のみ受ける

$$p(w_n | w_1, \dots, w_{n-1}) \approx p(w_n | w_{n-N+1}, \dots, w_{n-1})$$

- ユニグラム(0次のマルコフ過程)

$$p(w_n | w_1, \dots, w_{n-1}) \approx p(w_n)$$

- バイグラム(1次のマルコフ過程)

$$p(w_n | w_1, \dots, w_{n-1}) \approx p(w_n | w_{n-1})$$

- トライグラム(2次のマルコフ過程)

$$p(w_n | w_1, \dots, w_{n-1}) \approx p(w_n | w_{n-2}, w_{n-1})$$



高次のマルコフ過程の問題

- 高い次数のマルコフ過程
 - より精度が高いことが期待できる
 - 訓練データに出現しなかった単語列に対しては推定ができない(ゼロ頻度問題、データスパースネス)
 - 次数が高いほどデータスパースになってしまふ



スムージング

- 線形補間法(linear interpolation)
 - Nグラムの確率値を低次のMグラム($M < N$)の確率値と線形に補間する方法

$$\tilde{p}(w_n | w_{n-N+1}, \dots, w_{n-1}) \equiv \lambda p(w_n | w_{n-N+1}, \dots, w_{n-1}) + (1-\lambda) p(w_n | w_{n-N+2}, \dots, w_{n-1})$$

- トライグラムの場合

$$\tilde{p}(w_n | w_{n-2}, w_{n-1}) \equiv \lambda_1 p(w_n | w_{n-2}, w_{n-1}) + \lambda_2 p(w_n | w_{n-1}) + \lambda_3 p(w_n)$$

ただし

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$



スムージング

- 補間係数 λ を推定
 - ヘルドアウト補間法(held-out interpolation)
 - 訓練データとヘルドアウトデータにわけ
 - 訓練データでNグラムを学習
 - ヘルドアウトデータで補間係数を学習
 - 補間係数はEMアルゴリズムで最尤推定
 - トライグラムの場合

$$\tilde{\lambda}_1 = \frac{1}{|D_h|} \sum_{n=1}^{|D_h|} \frac{\lambda_1 p(w_n | w_{n-2}, w_{n-1})}{\lambda_1 p(w_n | w_{n-2}, w_{n-1}) + \lambda_2 p(w_n | w_{n-1}) + \lambda_3 p(w_n)}$$

$$\tilde{\lambda}_2 = \frac{1}{|D_h|} \sum_{n=1}^{|D_h|} \frac{\lambda_2 p(w_n | w_{n-1})}{\lambda_1 p(w_n | w_{n-2}, w_{n-1}) + \lambda_2 p(w_n | w_{n-1}) + \lambda_3 p(w_n)}$$

$$\tilde{\lambda}_3 = \frac{1}{|D_h|} \sum_{n=1}^{|D_h|} \frac{\lambda_3 p(w_n)}{\lambda_1 p(w_n | w_{n-2}, w_{n-1}) + \lambda_2 p(w_n | w_{n-1}) + \lambda_3 p(w_n)}$$



スムージング

- 削除補間法(deleted interpolation)
 - データを m 個に分割(L_1, L_2, \dots, L_m)
 - L_2, \dots, L_m を N グラムの訓練データ、 L_1 を補間係数推定のためのヘルドアウトデータとしてヘルドアウト推定法で学習
 - 同様に L_1, L_3, \dots, L_m で N グラム、 L_2 で補間係数を学習
 - ...
 - これを繰り返して、 L_1, \dots, L_{m-1} で N グラム、 L_m で補間係数を学習
 - 以上のようにして求めた補間係数の平均をとる
 - $|L_i|=1$ の時、リーヴィング・ワン・アウト法と呼ばれる（リーブ・ワン・アウトともいう）



COLLINS MODEL 1

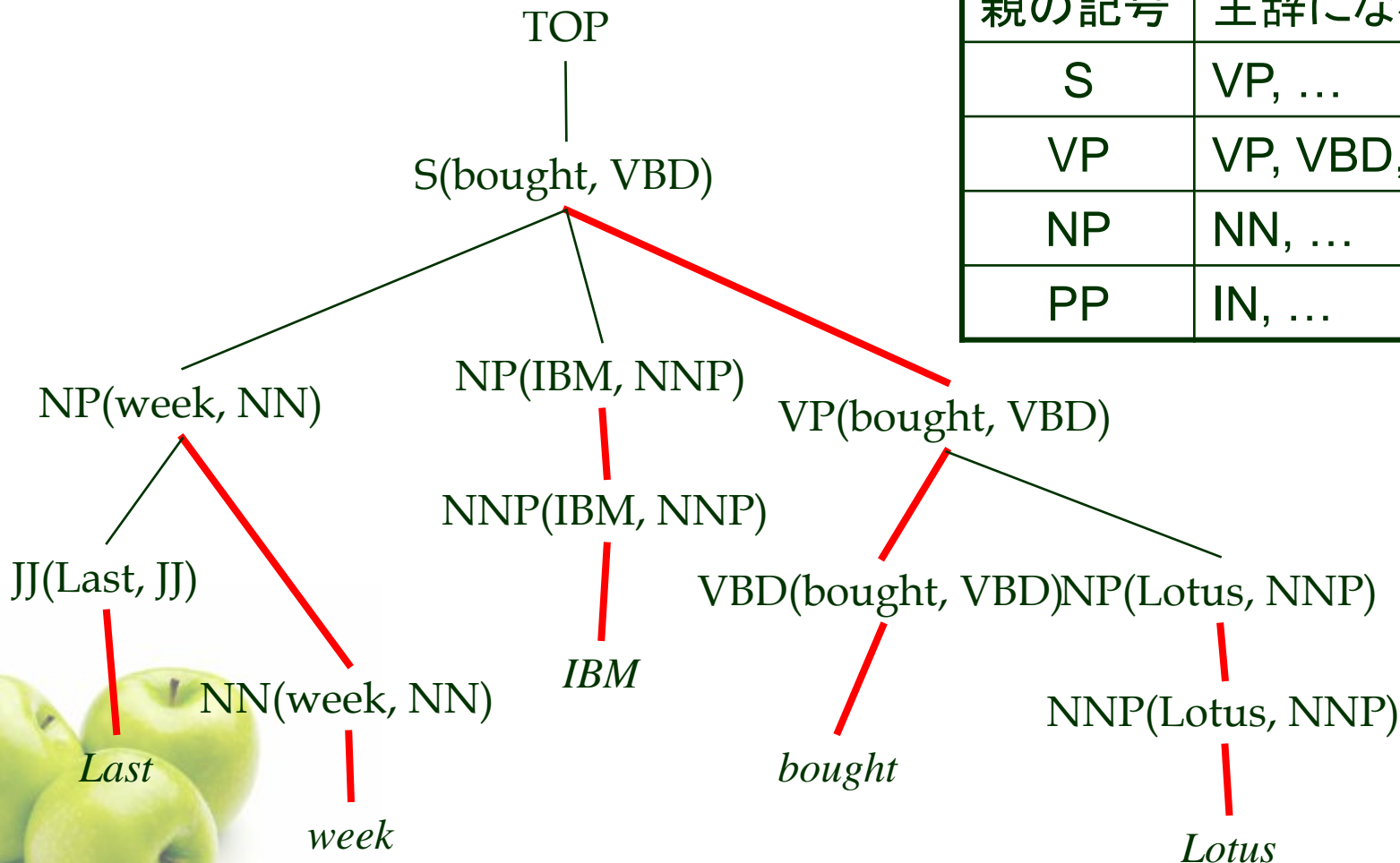


ツリーバンクの変形

- 語彙化

Head percolation table

親の記号	主辞になる子の記号
S	VP, ...
VP	VP, VBD, VBZ, ...
NP	NN, ...
PP	IN, ...



語彙化

- 構文木ノードが非終端記号から(非終端記号 + 主辞語 + 主辞品詞)の組になった
 - pure CFGの構文木ノード: NP
 - 語彙化 CFGの構文木ノード: NP(IBM, NNP)
- 書換規則も語彙化されたノードで表現
 - pure CFG: $S \rightarrow NP \ NP \ VP$
 - 語彙化CFG: $S(\text{bought}, \text{VBD}) \rightarrow NP(\text{week}, \text{NN})$
 $NP(\text{IBM}, \text{NNP}) \ VP(\text{bought}, \text{VBD})$



語彙化CFGの書換規則

- 書換規則

- $M(h) \rightarrow L_n(l_n) \dots L_1(l_1) H(h) R_1(r_1) \dots R_m(r_m)$

- H : head-child

- M : 親

- L : 主辞の左側の修飾句

- R : 主辞の右側の修飾句

- h : 主辞語と主辞品詞のペア

- l_i : L_i の主辞語と主辞品詞のペア

- r_i : R_i の主辞語と主辞品詞のペア

- 書換規則の確率

$$p(L_{n+1}(l_{n+1}), \dots, L_1(l_1), H(h), R_1(r_1), \dots, R_{m+1}(r_{m+1}) \mid M, h)$$

書換規則のマルコフ化

- 書換規則の確率

$$p(L_{n+1}(l_{n+1}), \dots, L_1(l_1), H(h), R_1(r_1), \dots, R_{m+1}(r_{m+1}) \mid M, h) =$$

$$p_h(H \mid M, h) \times$$

$$\prod_{i=1}^{n+1} p_l(L_i(l_i) \mid L_1(l_1), \dots, L_{i-1}(l_{i-1}), M, h, H) \times$$

$$\prod_{i=1}^{m+1} p_r(R_i(r_i) \mid L_1(l_1), \dots, L_{n+1}(l_{n+1}), R_1(r_1), \dots, R_{i-1}(r_{i-1}), M, h, H)$$

ただし

$$L_{n+1}(l_{n+1}) = STOP, R_{m+1}(r_{m+1}) = STOP$$

書換規則のマルコフ化

- 書換規則の確率

- 0次のマルコフ過程 (c.f. charniak(2000)は3次のマルコフ文法)

$$p_l(L_i(l_i) | L_1(l_1), \dots, L_{i-1}(l_{i-1}), M, h, H) \approx p_l(L_i(l_i) | M, h, H)$$

$$p_r(R_i(r_i) | L_1(l_1), \dots, L_{n+1}(l_{n+1}), R_1(r_1), \dots, R_{i-1}(r_{i-1}), M, h, H) \approx p_r(R_i(r_i) | M, h, H)$$

- つまり、

$$p(L_{n+1}(l_{n+1}), \dots, L_1(l_1), H(h), R_1(r_1), \dots, R_{m+1}(r_{m+1}) | M, h) =$$

$$p_h(H | M, h) \times \prod_{i=1}^{n+1} p_l(L_i(l_i) | M, h, H) \times \prod_{i=1}^{m+1} p_r(R_i(r_i) | M, h, H)$$

(語彙化と) 書換規則の確率の定義が変わるだけで、その他はpure PCFGと同じであることに注意！

書換規則のマルコフ化: 例

- $S(\text{bought}) \rightarrow NP(\text{week}) NP(\text{IBM}) VP(\text{bought})$
- $p(NP(\text{week})NP(\text{IBM})VP(\text{bought}) | S(\text{bought})) =$
 $p_h(VP | S, \text{bought}) \times$
 $p_l(NP(\text{IBM}) | S, VP, \text{bought}) \times$
 $p_l(NP(\text{week}) | S, VP, \text{bought}) \times$
 $p_l(\text{STOP} | S, VP, \text{bought}) \times$
 $p_r(\text{STOP} | S, VP, \text{bought})$



Distance関数の追加

- マルコフ化の際にDistance関数を追加

$$p_l(L_i(l_i) | L_1(l_1), \dots, L_{i-1}(l_{i-1}), M, h, H) \approx p_l(L_i(l_i) | M, h, H, \delta_l(i-1))$$

$$p_r(R_i(r_i) | L_1(l_1), \dots, L_{n+1}(l_{n+1}), R_1(r_1), \dots, R_{i-1}(r_{i-1}), M, h, H) \approx$$

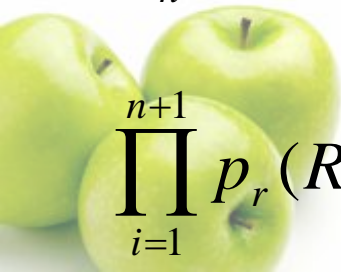
$$p_r(R_i(r_i) | M, h, H, \delta_r(i-1))$$

- 書換規則の確率

$$p(L_{n+1}(l_{n+1}), \dots, L_1(l_1), H(h), R_1(r_1), \dots, R_{m+1}(r_{m+1}) | M, h) =$$

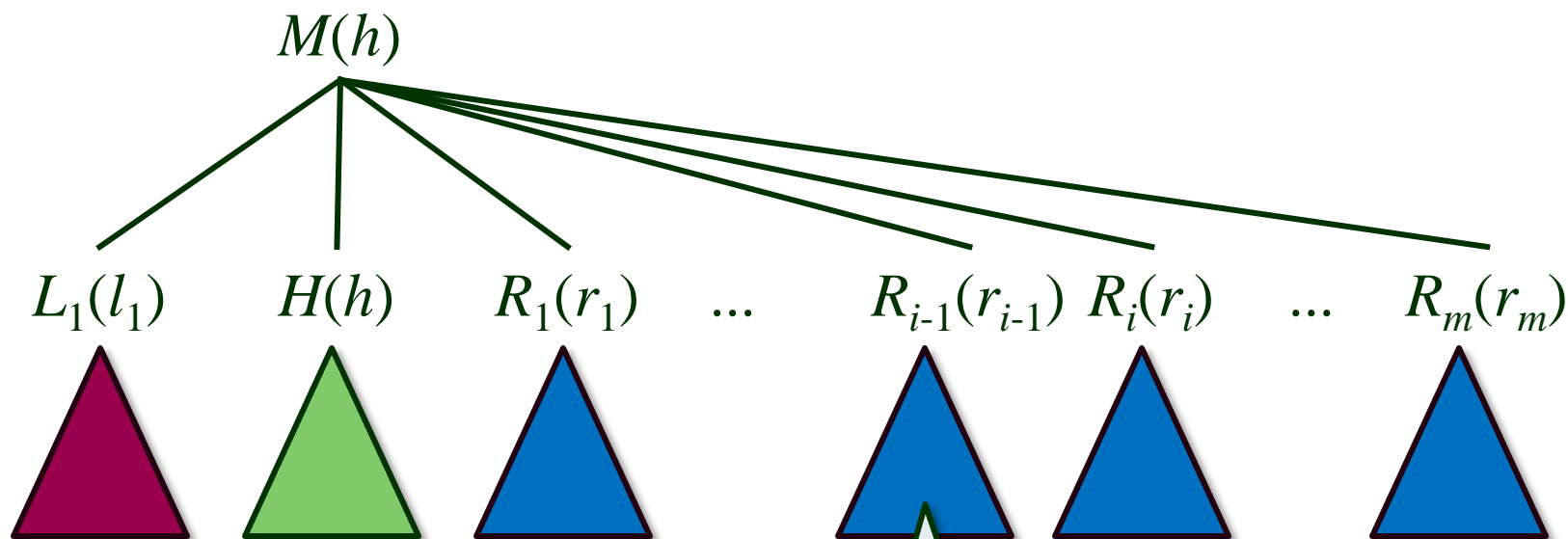
$$p_h(H | M, h) \times \prod_{i=1}^{n+1} p_l(L_i(l_i) | M, h, H, \delta_l(i-1)) \times$$

$$\prod_{i=1}^{n+1} p_r(R_i(r_i) | M, h, H, \delta_r(i-1))$$



Distance関数の中身

- Distance関数が指す部分木



$\delta(i-1)$ はこの木のこと



Distance関数の中身

- 求めようとする構文木ノードの条件付き確率 $p(R_i(r_i) | \dots)$ の一つ手前のノード $R_{i-1}(r_{i-1})$ の下に広がる部分構文木が対象
- Distance関数の返す値
 - R_1 か L_1 である？
 - 部分構文木の下に動詞を含むか否か？

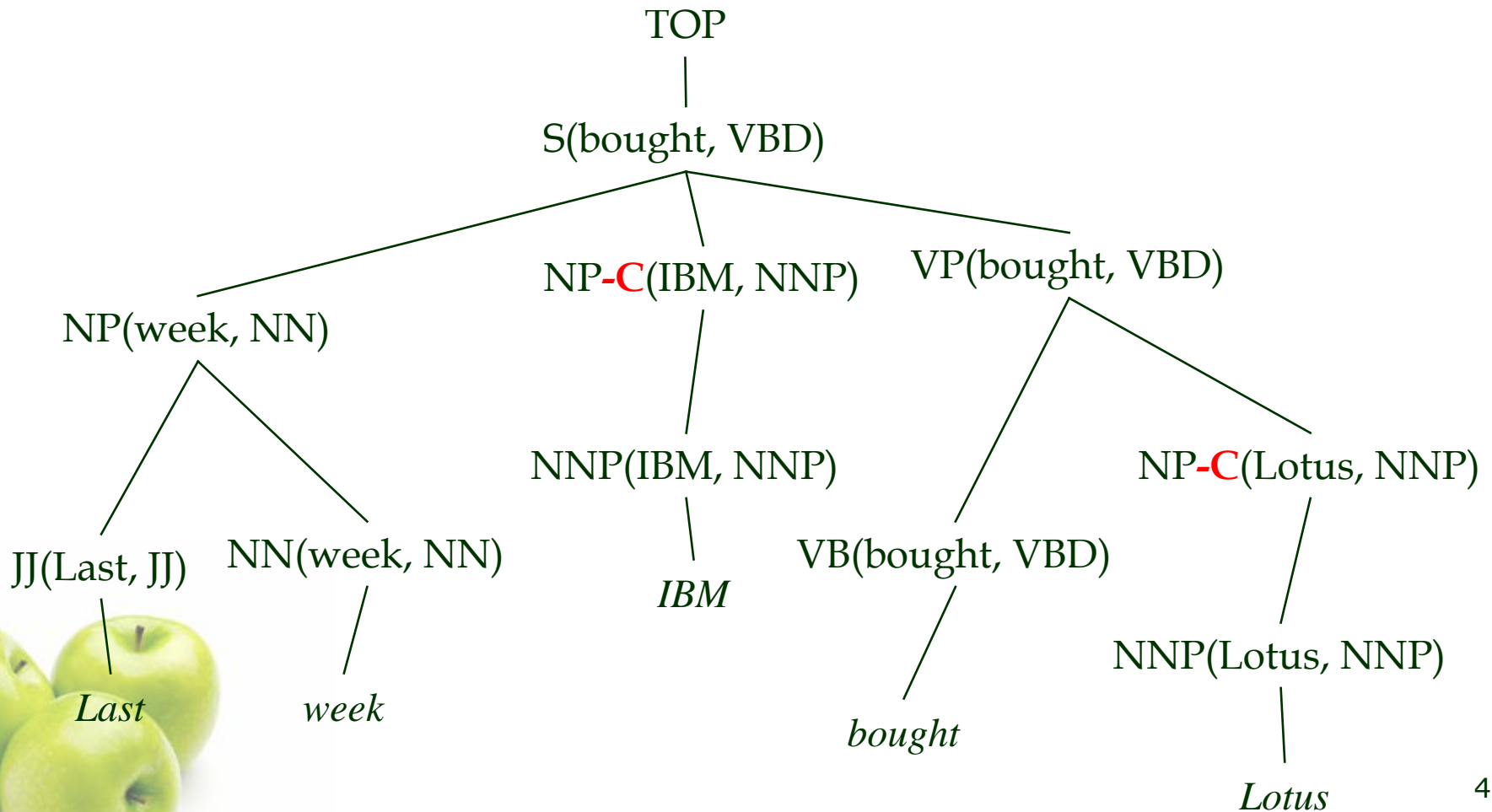


COLLINS MODEL 2



ツリーバンクの変形

- 語彙化+補語(句)/修飾語(句)の区別



ツリーバンクの変形

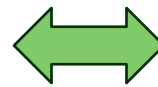
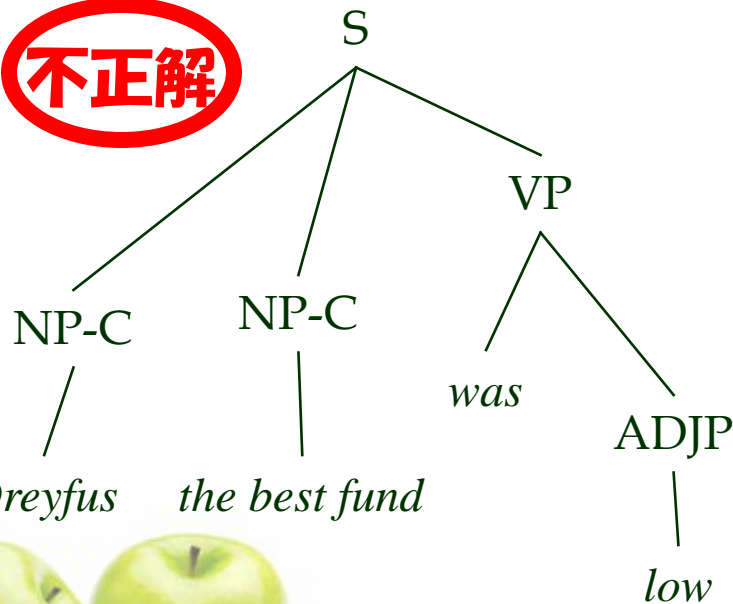
- 補語(complements)/修飾語(adjuncts)の区別
 - 次の条件を満たす非終端記号は全て補語(complements)
 - 非終端記号は次のうちのいずれかである
 - (1) 親がSであるNP, SBAR, S
 - (2) 親がVPであるNP, SBAR, S
 - (3) 親がSBARであるS
 - 非終端記号は次のうちのいずれかのsemantic tagをもっていない
 - ADV, VOC, BNF, DIR, EXT, LOC, NMR, TMP, CLR or PRP
 - それに加えて、前置詞句の主辞の後にすぐに続く兄弟ノードは補語である



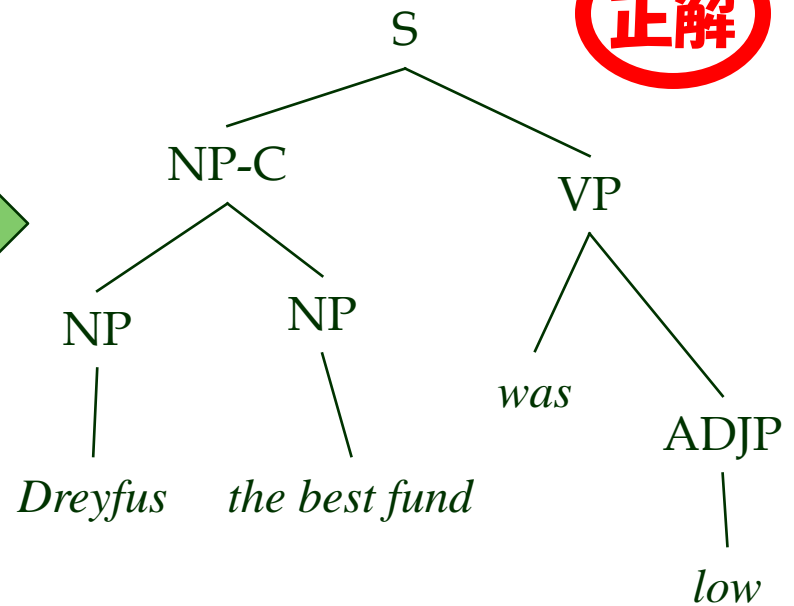
下位範疇化フレーム: 問題

- 補語や修飾語の区別をつけても、文法的に誤った構文と正しい構文の確率に差がつかない
- (例1)

不正解



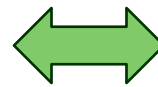
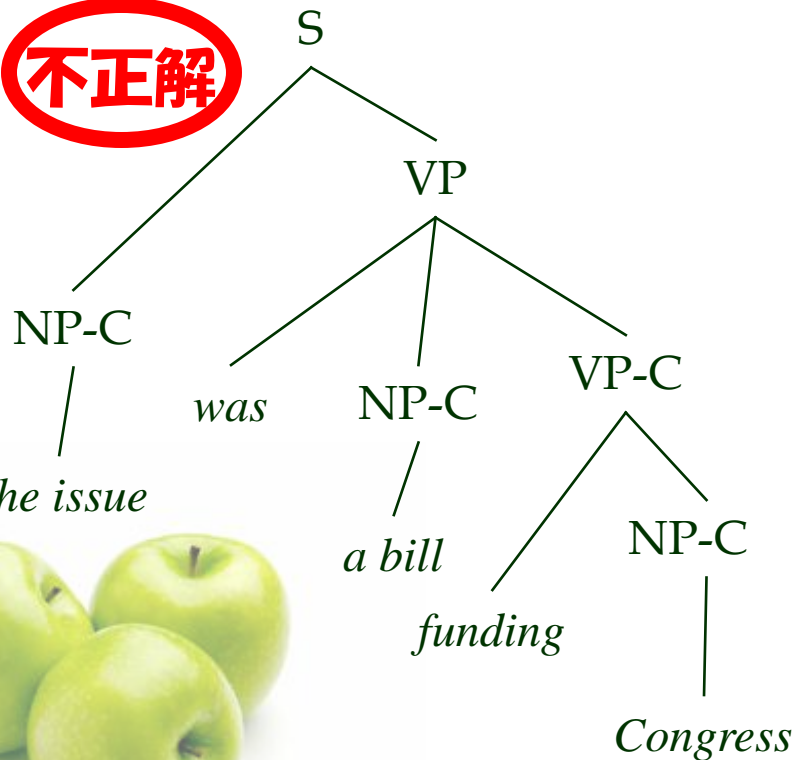
正解



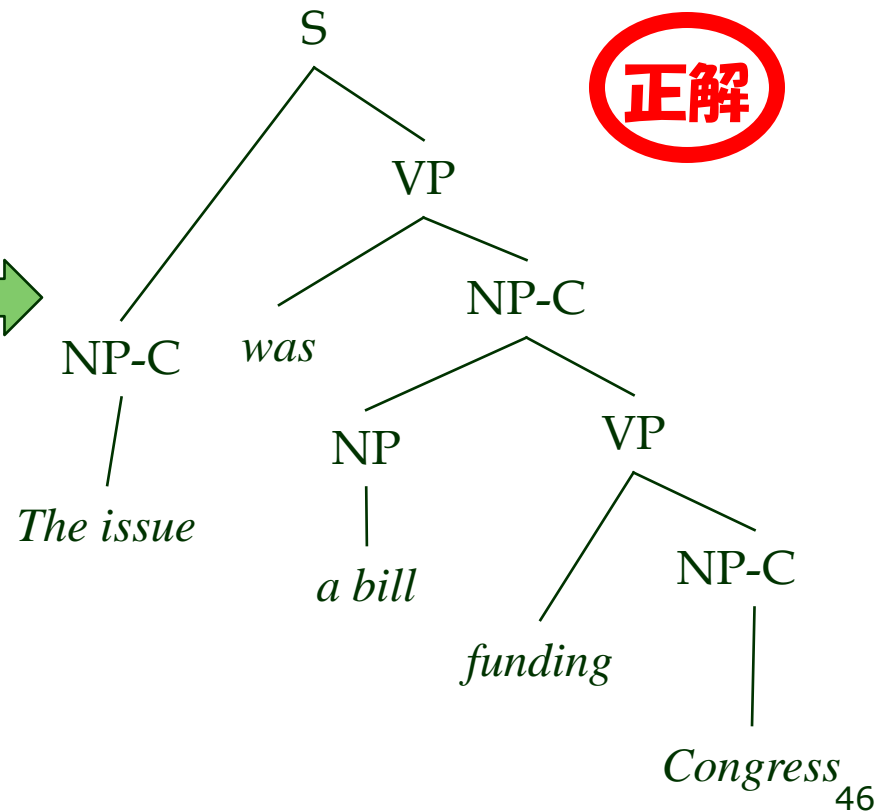
下位範疇化フレーム: 問題

- 補語や修飾語の区別をつけても、文法的に誤った構文と正しい構文の確率に差がつかない
- (例2)

不正解



正解



下位範疇化フレーム

- 解決策

- 下位範疇化フレーム(subcat frame)を導入
 - 補語として取る非終端記号の多重集合(multi set)
 - HPSGで出てきたVALの下のSUBJやCOMPSやSPR



書換規則の確率モデル

- LC(左側の下位範疇化フレーム)とRC(右側の下位範疇化フレーム)の導入

$$\begin{aligned} p(L_{n+1}(l_{n+1}), \dots, L_1(l_1), H(h), R_1(r_1), \dots, R_{m+1}(r_{m+1}) | M, h) = \\ p_h(H | M, h) \times p_{lc}(LC | M, H, h) \times p_{rc}(RC | M, H, h) \times \\ \prod_{i=1}^{n+1} p_l(L_i(l_i) | M, h, H, \delta_l(i-1), LC) \times \\ \prod_{i=1}^{m+1} p_r(R_i(r_i) | M, h, H, \delta_r(i-1), RC) \end{aligned}$$

- 例

- RC={NP-C, NP-C}, LC={NP-C}

- $p_{lc}(\{\text{NP-C, NP-C} | S, VP, \text{bought}\})$ や $p_{rc}(\{\text{NP-C, VP-C} | VP, VB, \text{was}\})$ は低い

下位範疇化フレーム

- 下位範疇化フレームは補語をひとつとる度に消費される（減っていく）
- 下位範疇化フレームに要素が残っている間はSTOPに対する確率は0
- 下位範疇化フレームに要素が無い場合は、補語をとる確率は0



下位範疇化フレーム: 例

- S(bought) → NP(week) NP-C(IBM) VP(bought)の確率

$$p_h(\text{VP} | \text{S}, \text{bought}) \times$$

$$p_{lc}(\{\text{NP-C}\} | \text{S}, \text{VP}, \text{bought}) \times$$

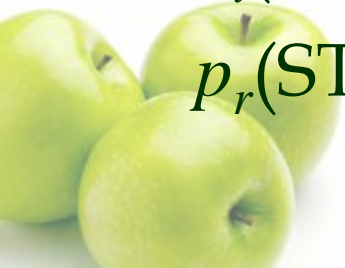
$$p_{rc}(\{\} | \text{S}, \text{VP}, \text{bought}) \times$$

$$p_l(\text{NP-C(IBM)} | \text{S}, \text{VP}, \text{bought}, \{\text{NP-C}\}) \times$$

$$p_l(\text{NP(week)} | \text{S}, \text{VP}, \text{bought}, \{\}) \times$$

$$p_l(\text{STOP} | \text{S}, \text{VP}, \text{bought}, \{\}) \times$$

$$p_r(\text{STOP} | \text{S}, \text{VP}, \text{bought}, \{\})$$



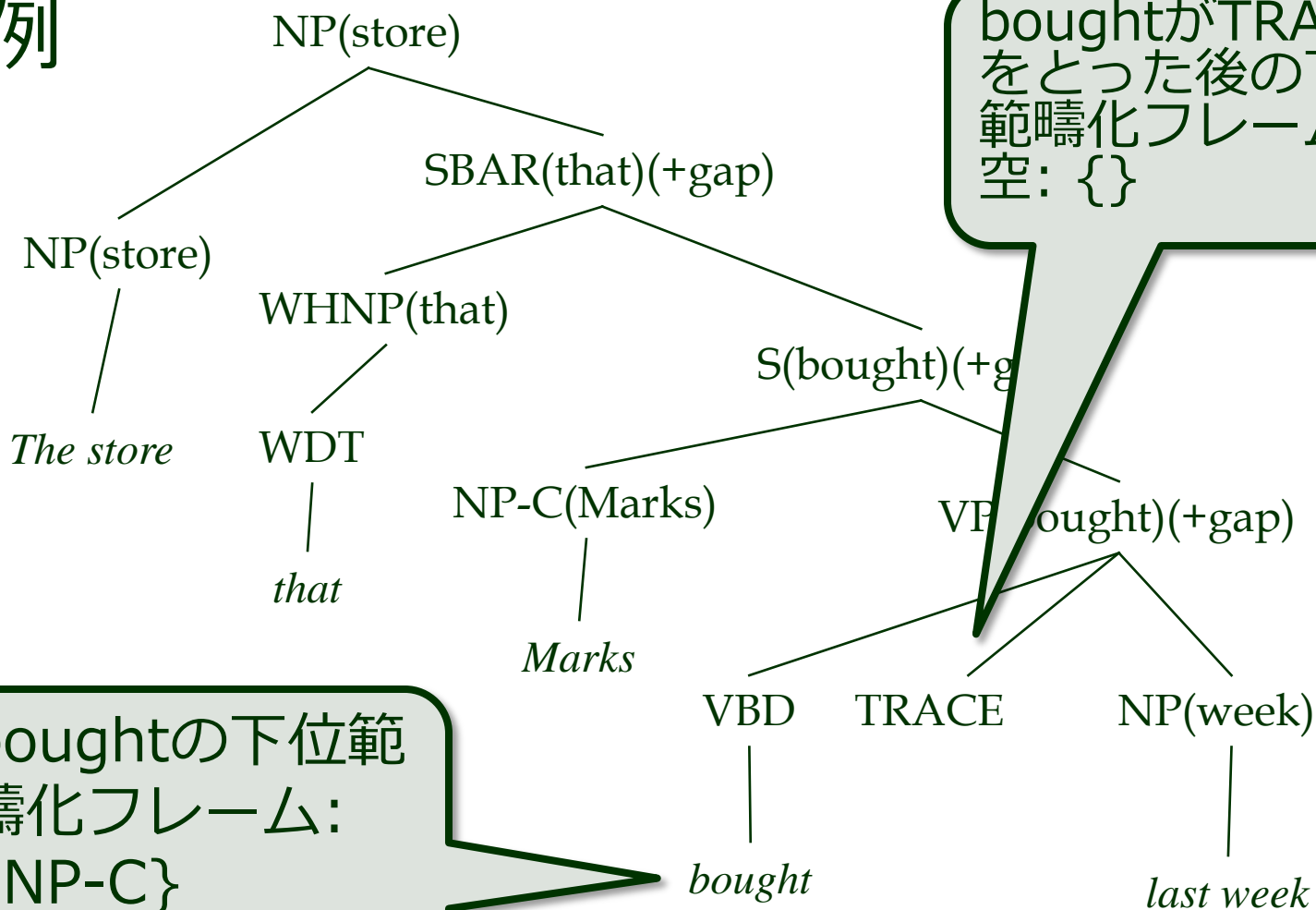
Collins Model 3

- 痕跡(Trace)とwh-移動
 - ex.1 The store (SBAR which TRACE bought Brooks Brothers)
 - ex.2 The store (SBAR which Marks bought TRACE)
 - ex.3 The store (SBAR which Marks bought Brooks Brothers from TRACE)



Collins Model 3

- 例



boughtがTRACE
をとった後の下位
範疇化フレームは
空: {}

boughtの下位範
疇化フレーム:
{NP-C}

スムージング

主辞品詞

主辞語

- $p_l \subset p$

$$p_l(L_i(\langle lt_i, lw_i \rangle) | M, h, H, \delta_l(i-1), LC) = p_{l1}(L_i(lt_i) | M, h, H, \delta_l(i-1), LC) \times p_{l2}(lw_i | M, h, H, \delta_l(i-1), LC)$$

$$p_r(R_i(\langle rt_i, rw_i \rangle) | M, h, H, \delta_r(i-1), RC) = p_r(R_i(rt_i) | M, h, H, \delta_r(i-1), RC) \times p_r(rw_i | M, h, H, \delta_r(i-1), RC)$$

- 削除補間法(deleted interpolation)

back-off level	$p_h(H ...)$	$p_{lc}(LC ...)$ $p_{rc}(RC ...)$	$p_{l1}(L_i(lt_i) ...)$ $p_{r1}(R_i(rt_i) ...)$	$p_{l2}(lw_i ...)$ $p_{r2}(rw_i ...)$
1	M, w, t	M, H, w, t	M, H, w, t, δ, LC	$L_i, lt_i, M, H, w, t, \delta, LC$
2	M, t	M, H, t	M, H, t, δ, LC	$L_i, lt_i, M, H, t, \delta, LC$
3	M	M, H	M, H, δ, LC	L_i, lt_i
4	-	-	-	lt_i

性能評価

- PARSEVALという基準で評価
 - 構文木ノードの位置（左端の位置と右端の位置）と非終端記号ラベル
 - Labeled Precision (LP)=(パーザーが正解したラベル数)/(パーザーが出力したラベル数)
 - Labeled Recall (LR) = (パーザーが正解したラベル数)/(ツリーバンク中のラベル数)
 - Labeled F-Score(F1-Score) = labeled precisionとlabeled recallの調和平均= $2*LP*LR/(LP+LR)$

model	LP	LR	LF
Magerman95	84.3%	84.0%	84.1%
Collins96	85.7%	85.3%	85.5%
Model 1	87.6%	86.8%	87.2%
Model 2	88.1%	87.5%	87.8%
Model 3	88.1%	87.5%	87.8%

まとめ

- マルコフ文法
 - マルコフ過程
 - スムージング
 - Collins Model 1
 - Collins Model 2
 - Collins Model 3
- 講義資料
 - <http://aiweb.cs.ehime-u.ac.jp/~ninomiya/ai2/>

