

教師なし構文解析の進展

最先端構文解析とその周辺

統計数理研究所 2012.12.19

能地 宏

東京大学大学院 情報理工学系研究科

修士2年

今日の話

- ▶ 教師なし構文解析とは何か
 - 問題の意義
 - 何故教師なしで考えるのか？

- ▶ 教師なし係り受け解析における代表的なモデルの紹介（次ページ）
 - モデルの考え方
 - 現状，どれぐらいうまくいくのか
 - 今後の展望

モデル化の方針

- ▶ PCFGに変換した上で、手を加えていく
 - Klein & Manning (2002)
 - Dependency Model with Valence (Klein & Manning, 2004)
 - 拡張がたくさん
 - Extended Model
 - Lexicalized Model
 - Parameter Tying ⇒ Shared Logistic, Phylogenetic Model, Multilingual Setting, etc
- ▶ その他のモデル
 - Projective Tree を生成する別の生成モデル
 - Common Cover Link (Seginer 2007)

教師なし解析の目的

▶ 工学的目的

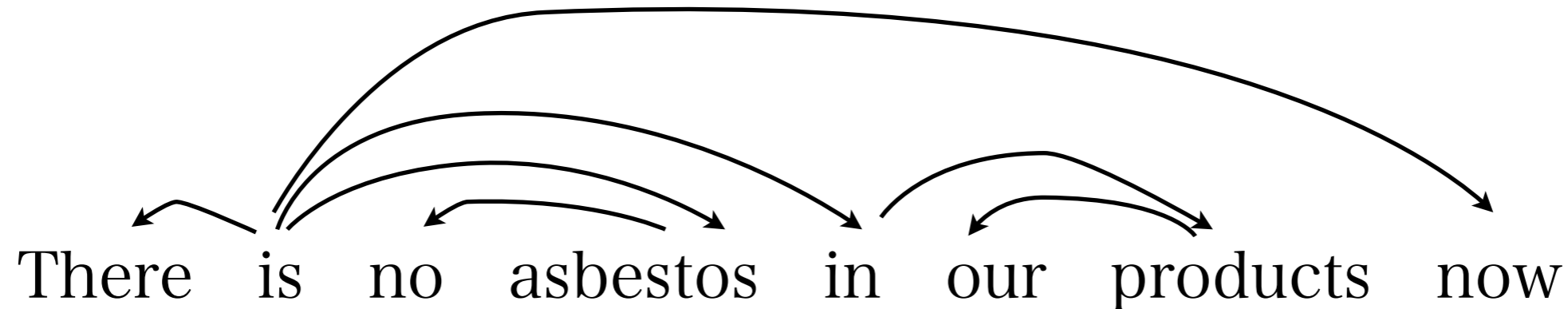
- 現在，構文解析を行うためには，ツリーバンクなどの，教師データが必要
- しかし，ツリーバンクの存在する，言語/ドメインは限られる
 - ツリーバンクの存在しない言語の解析
 - ツイッターなどの新しいドメインの解析をどうするか？
- 半教師あり学習をするにしても，教師なしの性能の良い生成モデルを定義しておくことは意味がある

▶ 科学的目的

- 人間が言語を獲得する仕組みは何も明らかにされていない
- 教師なし解析がうまくいけば，そこで用いたような情報を，赤ちゃんが使っているかもしれない，という手掛かりになる
- 言語をよく説明出来るモデル ⇒ 言語のある側面の本質を備えている

係り受け解析

- ▶ 単語同士の係り関係を導出する
- ▶ Treebankなどの人手で作った正解データと比較する
- ▶ headとargument
 - head : 部分木の中で, 最も重要な意味を表す単語
 - argument : headに付属する単語

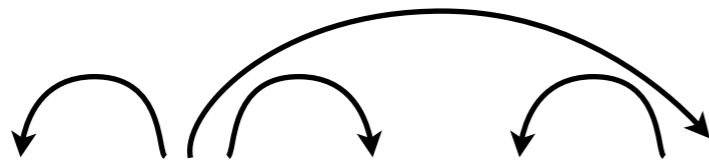


この構造を, 正解データを用いずに導出できるか?

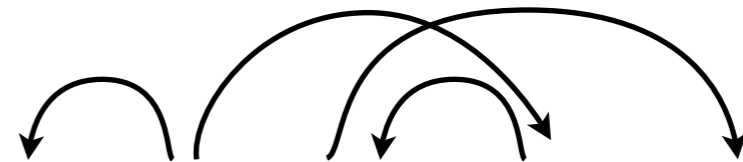
Projectivity

- ▶ 係り受け解析でよく用いられる仮定
- ▶ 文のなかで，係り関係が交差しない
- ▶ 多くの言語，多くの文で成り立つ性質
- ▶ 考える解空間の大きさを大きく狭めることが出来る
 - 今日紹介する多くのモデルで仮定されている

projective



Non-projective



問題設定自体がはらんでいる問題

- ▶ 係り関係は人間が定めたもの
- ▶ headとは何か，というのは言語学的に非常に難しい問題
- ▶ Treebank でも，言語によって正解の方針が異なる
 - 例：限定詞と名詞
 - The **country** での head は（通常）**country**
 - デンマーク語でのアノテーションは逆向き
 - 英語でも，The が head であるという主張もある (Abney, 1987)

とりあえずの策と，最終的な目標

- ▶ 色々問題は含んでいるが，言語学者のアノテーションは，ある種の本質を捉えている
- ▶ それを再現出来るようになることは，言語理解にとって重要といえる
- ▶ アプリケーションによる評価
 - 既存の教師ありによる解析よりも，ある種のアプリケーションでは，良い性能を示せるかもしれない（機械翻訳など）
 - ほとんど見ない ⇒ 精度が出ないから？
- ▶ 言語が統計的に処理出来るのであれば，それに従ってheadとは何かを人間の主観なしに定義することが出来るはず
- ▶ 教師なし学習の究極的な目標？

DMVまでの歴史

1992 Carroll & Charniak

PCFG & EM

extremely poor result

2001 Paskin

Grammatical bigram

2004 Klein & Manning

DMV

これ以降研究が盛んに…

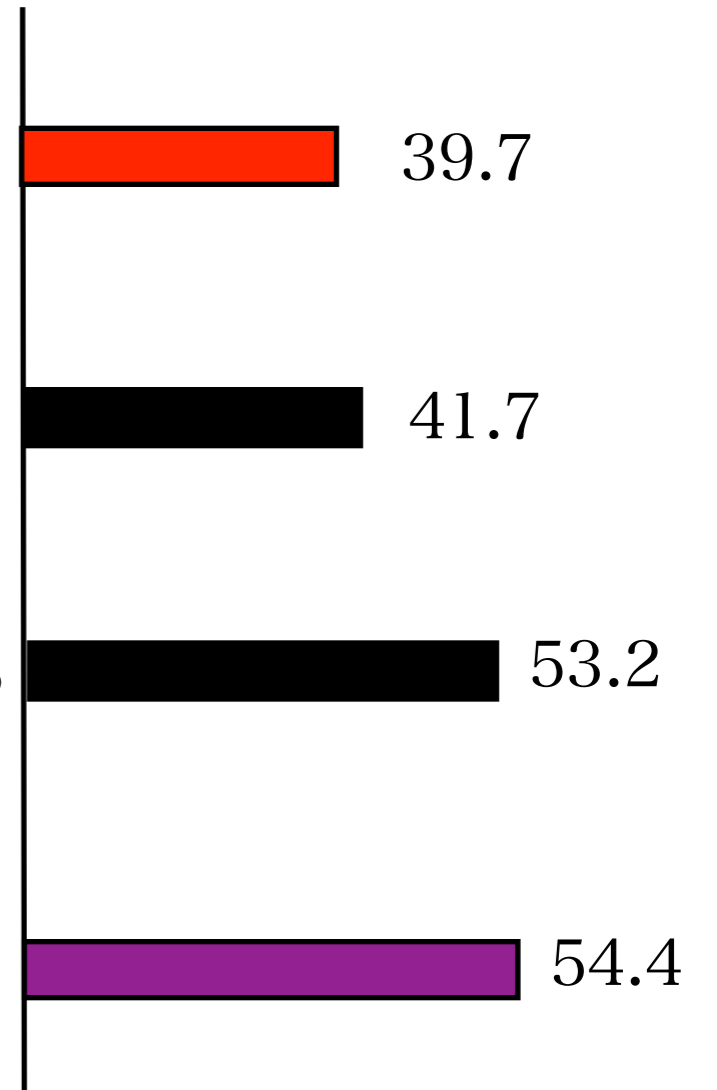
無向での評価

Paskin 39.7

ランダム 41.7

すべて隣にかける 53.2

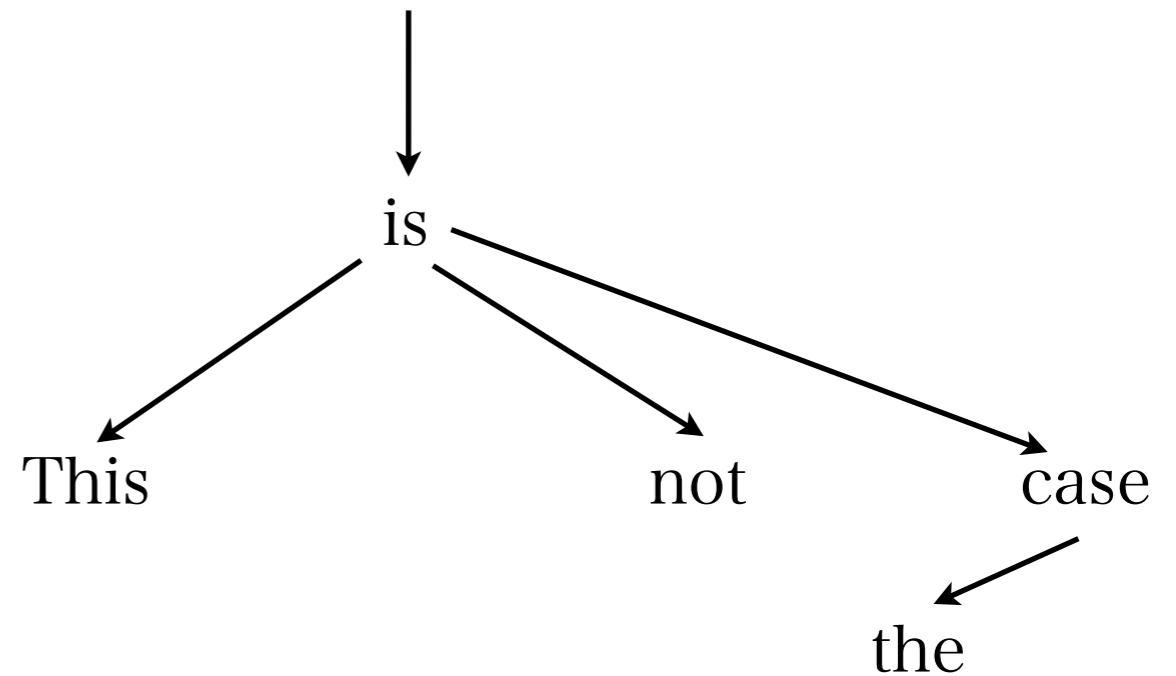
Klein & Manning 54.4



Grammatical bigram (Paskin, 2002)

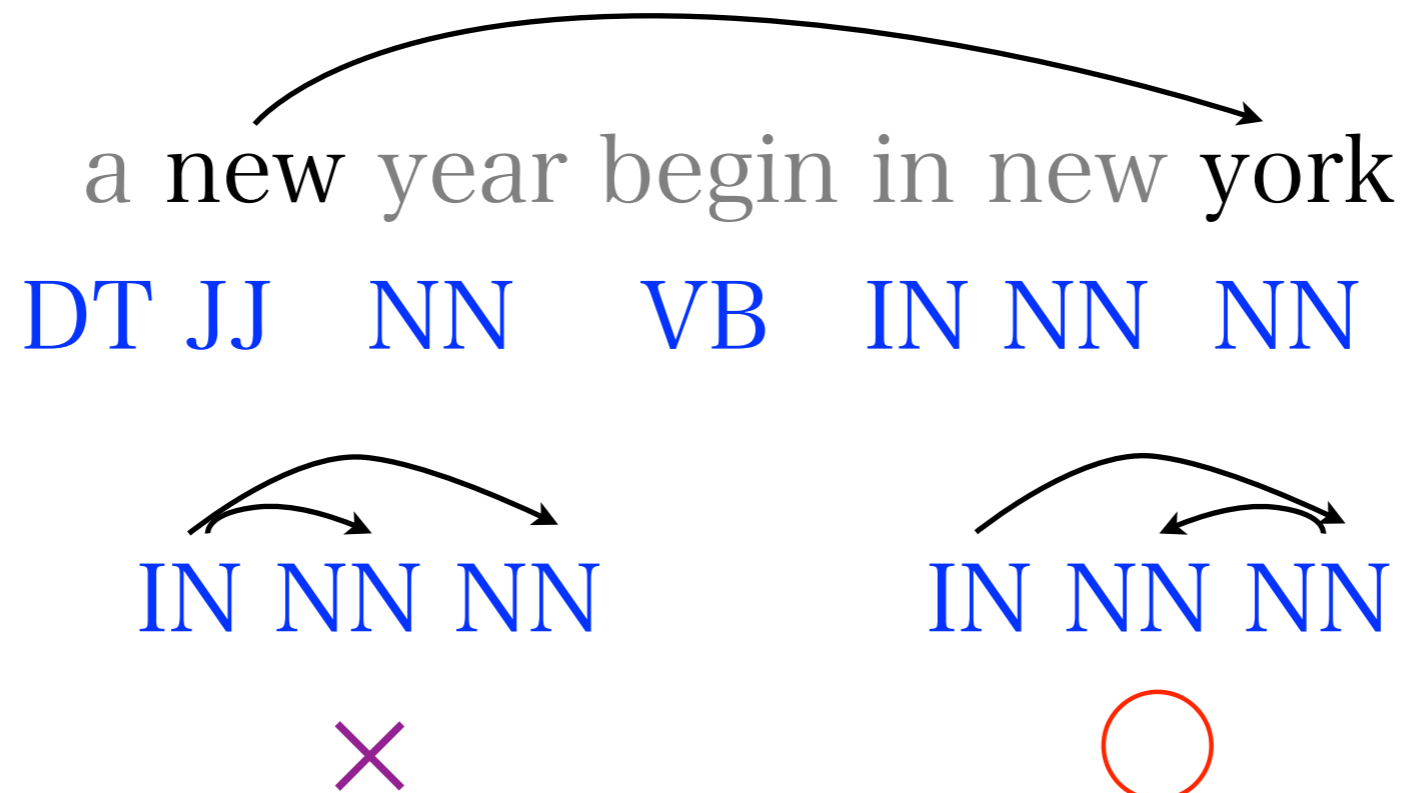
- ▶ Projective Tree に対する生成モデル
 - 木をランダムに生成 (一様分布)
 - 親ノードの単語から, 子ノードの単語が生成される
- ▶ 大量の生の文書 (300万文) から EM でパラメタを学習

$$\begin{aligned} p(G)p(\mathbf{w}|G) &\propto p(\mathbf{w}|G) \\ &= p(\text{is}|\text{root}) \times \\ &\quad p(\text{This}|\text{is}, \text{L}) \times \\ &\quad p(\text{not}|\text{is}, \text{R}) \times \\ &\quad p(\text{case}|\text{is}, \text{R}) \times \\ &\quad p(\text{not}|\text{case}, \text{L}) \end{aligned}$$



DMVはなぜうまくいったか

- ▶ (Paskin, 2001) の問題点
 - 木はランダムに生成されるため，共起しやすい単語同士が結びつく
- ▶ 単語同士の関係ではなく，品詞同士の関係をモデル化する
- ▶ valence = 各単語（品詞）の取りうるargumentの数をモデル化する
- ▶ Smart Initialization（これが結構重要）



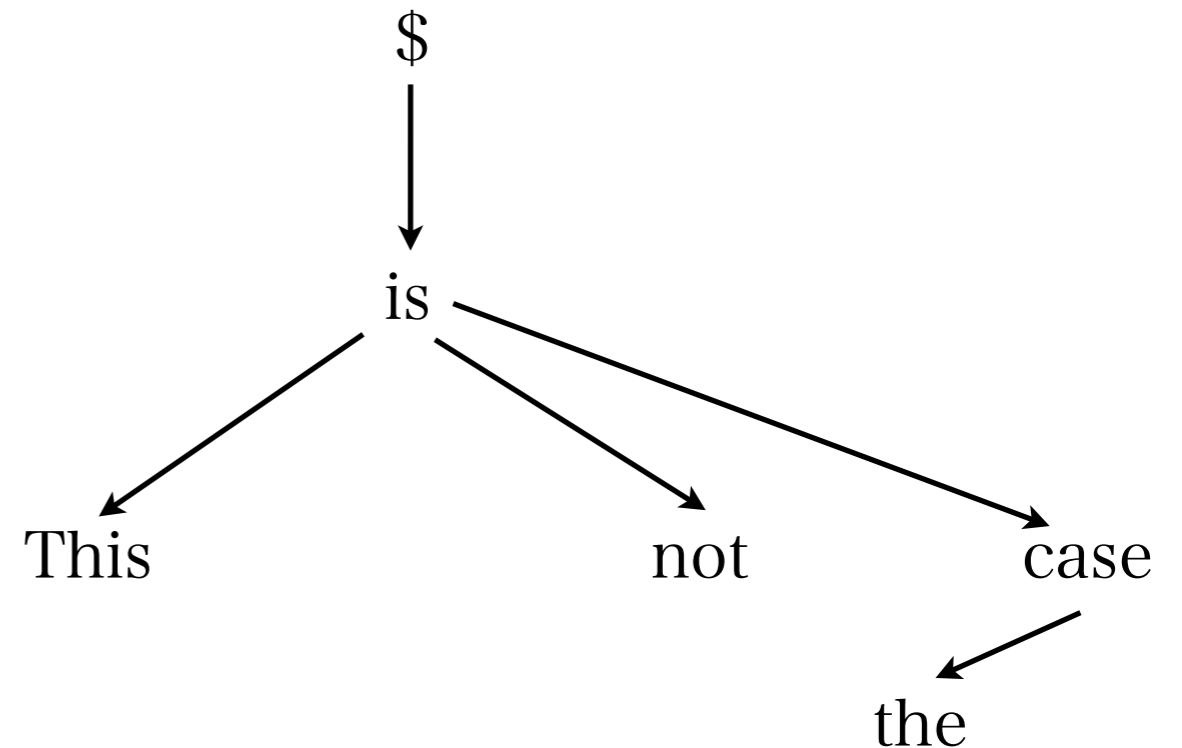
(Carroll & Charniak, 2001)

Dependency Model with Valence

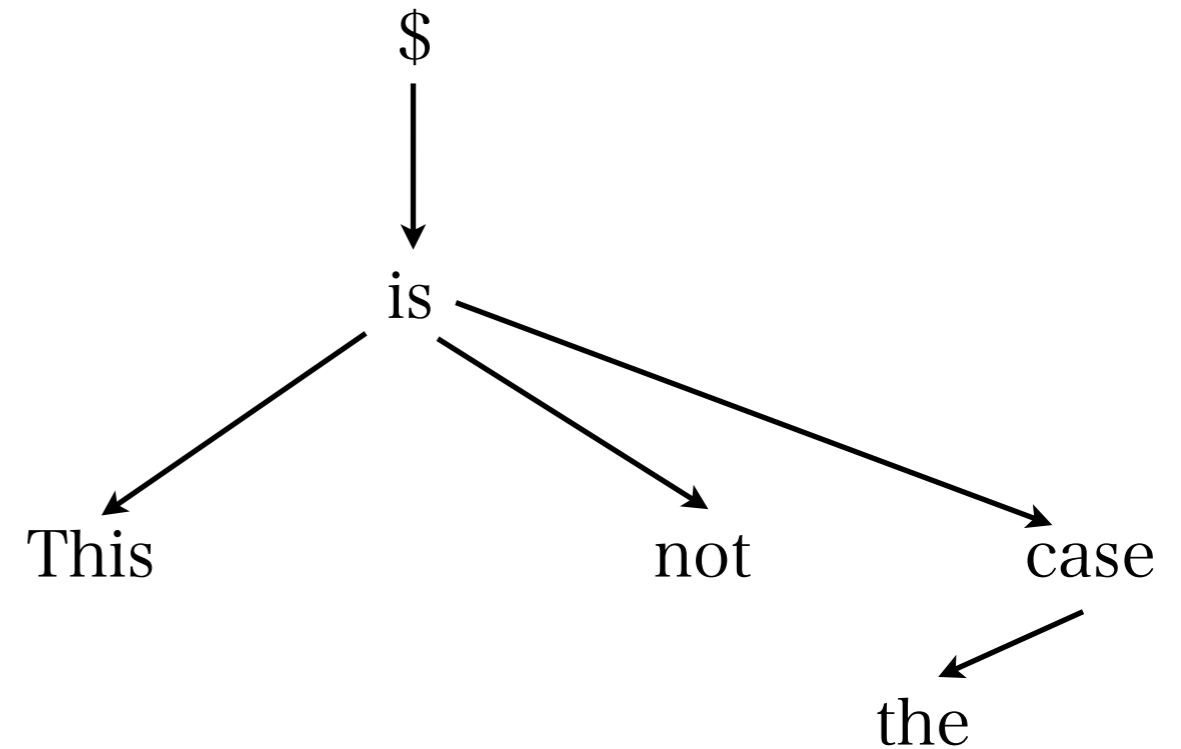
- ▶ 品詞の上で, Projectiveな係り受け関係を導出する生成モデル
 - 見やすくするため, 単語で説明
- ▶ Valence情報を組み込んでいる
 - 各品詞がargumentをいくつ取るか?

単語 w の下の部分木の確率 $p(\mathbf{y}_w | w)$

$$\begin{aligned} p(T) &= p(\mathbf{y}_\$ | \$) \\ &= p(\text{is} | \$) p(\mathbf{y}_{\text{is}} | \text{is}) \end{aligned}$$



$$\begin{aligned}
p(\mathbf{y}_{is}|is) = & p(\text{CONT}|\text{R}, is, v = 0) \times \\
& p(\text{case}|\text{R}, is)p(\mathbf{y}_{case}|\text{case}) \times \\
& p(\text{CONT}|\text{R}, is, v = 1) \times \\
& p(\text{not}|\text{R}, is)p(\mathbf{y}_{not}|\text{not}) \times \\
& p(\text{STOP}|\text{R}, is, v = 1) \\
& p(\text{CONT}|\text{L}, is, v = 0) \times \\
& p(\text{This}|\text{L}, is)p(\mathbf{y}_{\text{This}}|\text{This}) \times \\
& p(\text{STOP}|\text{L}, is, v = 1) \times
\end{aligned}$$



▶ Valence

- STOP or CONTINUE を決める際，各方向の最初の子供かどうかで異なる分布を用いる（外側から決まることに注意）

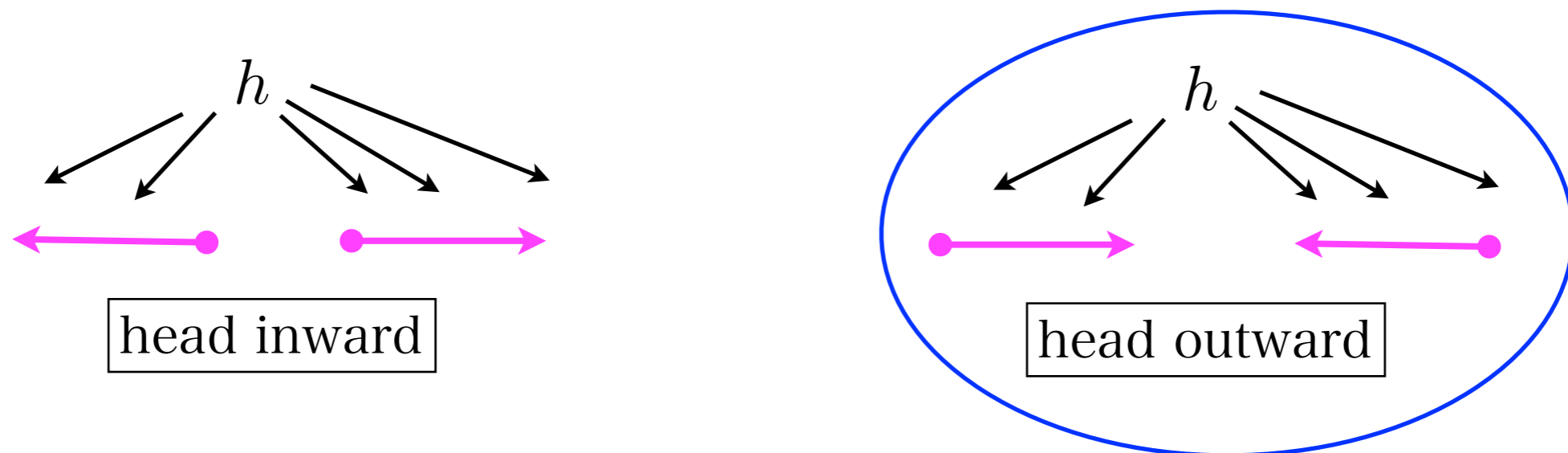
▶ Head automaton の一種 (Alshawi, 1996)

DMVのパラメタ

$p(w \$)$	文のheadとして w を生成
$p(\text{STOP} \text{R}, w, v = 0)$	右に子が いない 状態でSTOP
$p(\text{CONT} \text{R}, w, v = 0)$	右に子が いない 状態でCONTINUE
$p(\text{STOP} \text{R}, w, v = 1)$	右に子が いる 状態でSTOP
$p(\text{CONT} \text{R}, w, v = 1)$	右に子が いる 状態でCONTINUE
$p(w' \text{R}, w)$	右に単語 w' を生成
$p(\text{STOP} \text{L}, w, v = 0)$	左に子が いない 状態でSTOP
$p(\text{CONT} \text{L}, w, v = 0)$	左に子が いない 状態でCONTINUE
$p(\text{STOP} \text{L}, w, v = 1)$	左に子が いる 状態でSTOP
$p(\text{CONT} \text{L}, w, v = 1)$	左に子が いる 状態でCONTINUE
$p(w' \text{L}, w)$	左に単語 w' を生成

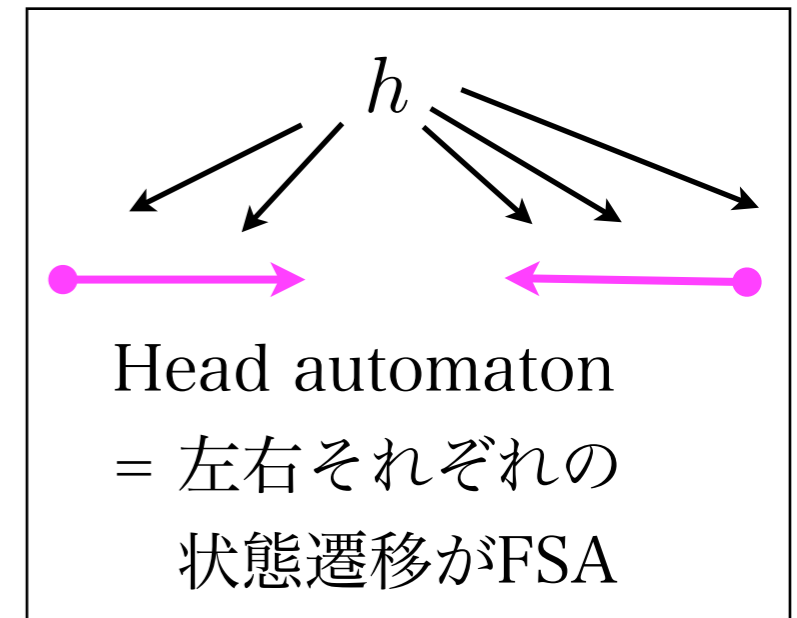
注意

- ▶ (Klein&Manning, 2004), (Klein, 2005), (Spitkovsky, 2010 etc)などは, 各 head に一番近い argument から, 順番に決めていく
- ▶ (Smith, 2006), (Cohen, 2008, 2009), (Headden III, 2009)などは, 各 head に最も遠い argument から順番に決めていく
- ▶ 両者に違いはないが, Extended Model (後述) では結果が異なる
- ▶ 最近の論文は後者で実装しているものが多いので, 以降 後者で説明



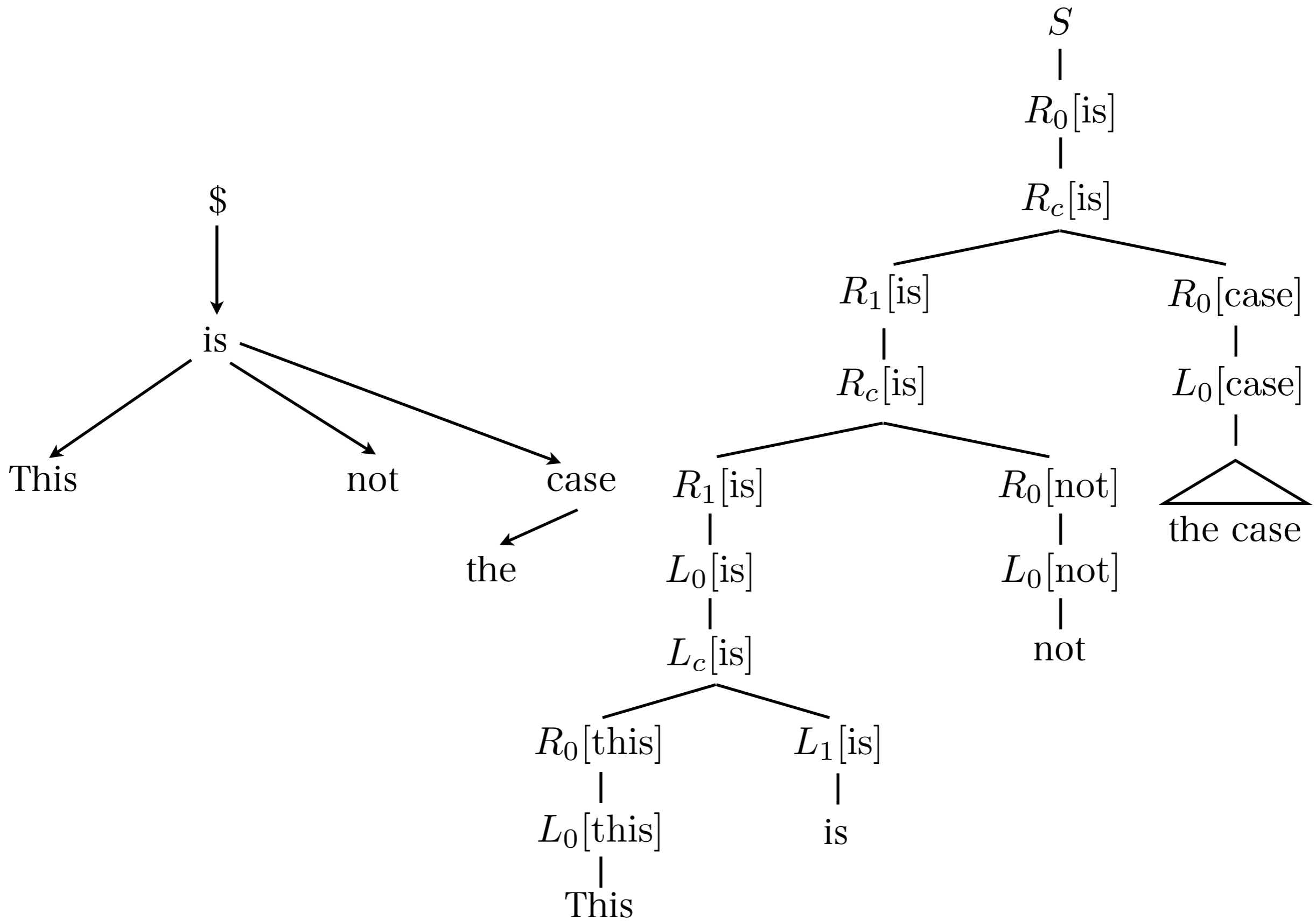
DMVにおけるValence

- ▶ 子を生成するときに，止まる，止まらないの選択を行う
 - (Collins, 1999) をシンプルしたもの
 - 子の数は，二項分布に従う
 - 少ない子の数が好まれる．その違いを，方向と，親の品詞に応じてモデル化



- ▶ 他のモデル化の方針は？
 - 例えば，子の数が多項分布に従う，としてはダメなのか
 - 生成モデルとしては考えられるが，Head automaton でなくなる
 - Head automatonをPCFGに変換することで，CYK / Inside Outside が使える

PCFGへの変換

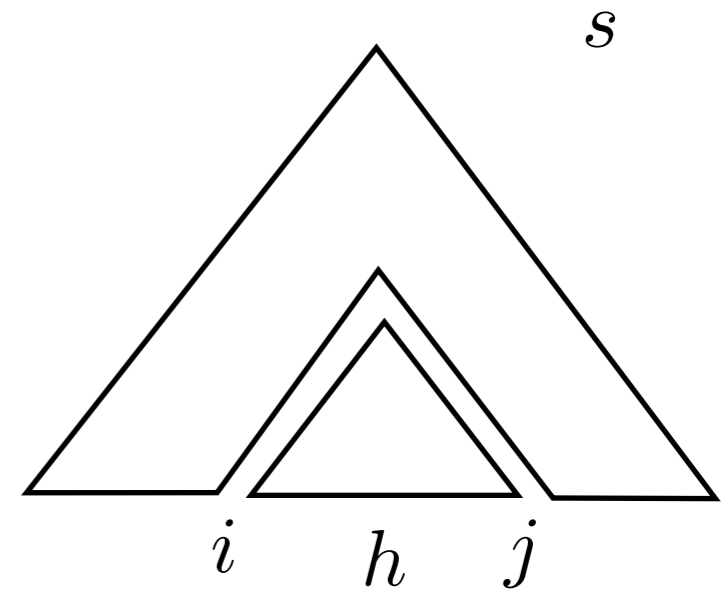


ルールとパラメタの対応

$p(w \$)$	$S \rightarrow R_0[w]$
$p(\text{STOP} \mathbf{R}, w, v = 0)$	$R_0[w] \rightarrow L_0[w]$
$p(\text{CONT} \mathbf{R}, w, v = 0)$	$R_0[w] \rightarrow R_c[w]$
$p(\text{STOP} \mathbf{R}, w, v = 1)$	$R_1[w] \rightarrow L_0[w]$
$p(\text{CONT} \mathbf{R}, w, v = 1)$	$R_1[w] \rightarrow R_c[w]$
$p(w' \mathbf{R}, w)$	$R_c[w] \rightarrow R_1[w]R_0[w]$
$p(\text{STOP} \mathbf{L}, w, v = 0)$	$L_0[w] \rightarrow w$
$p(\text{CONT} \mathbf{L}, w, v = 0)$	$L_0[w] \rightarrow L_c[w]$
$p(\text{STOP} \mathbf{L}, w, v = 1)$	$L_1[w] \rightarrow w$
$p(\text{CONT} \mathbf{L}, w, v = 1)$	$L_0[w] \rightarrow L_c[w]$
$p(w' \mathbf{L}, w)$	$L_c[w] \rightarrow R_0[w]L_1[w]$

パラメタの推定

- ▶ PCFGへ変換したので、一般的な推定法 (EM) で機械的にパラメタの推定が可能
- ▶ 各ルールとパラメタは、一対一に対応している
 - E-step : 各ルールの期待値を計算
 - M-step : 正規化して、新しいパラメタとする



$p(\text{STOP}|\mathbf{R}, \text{DT}, v = 0)$ の更新




$$p'(\text{STOP}|\mathbf{R}, \text{DT}, v = 0) \propto \sum_s c(\text{STOP}|\mathbf{R}, \text{DT}, v = 0, s)$$

$c(\text{STOP}|\mathbf{R}, \text{DT}, v = 0, s)$

$$= \frac{\sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \sum_{h=i:w_h=\text{DT}}^j \text{inside}(i, j, L_0, h) \text{outside}(i, j, R_0, h) p(\text{STOP}|\mathbf{R}, \text{DT}, v = 0)}{p(s)}$$

DMVの結果

- ▶ Penn Treebank WSJ から，10単語以下の文章を全て抜き出す
 - 推定されたパラメタを用いて，訓練データ自体を Viterbi パース
 - 精度 = (headを正しく判定出来た単語の数) / (全単語数)
- ▶ 短い文章のほうが解析が簡単なので，まずはこれを解析出来るモデルを目指す，という方針

RANDOM		30.1
R-ADJ		33.6
DMV		43.2

エラーの傾向

English using DMV			
Overproposals		Underproposals	
DT ← NN	3083	DT → NN	3079
NNP ← NNP	2108	NNP → NNP	1899
CC → ROOT	1003	IN ← NN	779
IN ← DT	858	DT → NNS	703
DT ← NNS	707	NN → VBZ	688
MD → VB	654	NN ← IN	669
DT → IN	567	MD ← VB	657
DT → VBD	553	NN → VBD	582
TO → VB	537	VBD ← NN	550
DT → VBZ	497	VBZ ← NN	543

- ▶ 矢印の向きを間違えることが多い
- ▶ しかし DT NN の head の位置については議論がある
- ▶ NNP NNP ⇒ 人名のhead は first / last name のどちら？

DMVの拡張

- ▶ DMVはとてもシンプルなモデル
 - 最低限の valence 情報を組み込んだモデル
- ▶ 08,09 年ぐらいから、様々な拡張が盛んに
 - (Cohen & Smith, 2008) : ベイズモデル, VBで推定
 - (Cohen & Smith, 2009) : Shared Logistic Normal
 - (Headden et al., 2009) : Extended Valence Grammar (EVG), Lexicalized EVG
 - (Blunsom & Cohn, 2010) : TSG + DMV
 - (Gillenwater et al., 2010) : Sparsity constrained Model
 - (Tu & Honavar, 2012) : Unambiguity resolution

Extended Valence Grammar

- ▶ PCFGを拡張することで、より豊富な情報を捉えよう、という方針
- ▶ DMVではargumentの生成に関して、順番を考慮しない

the big hungry dog



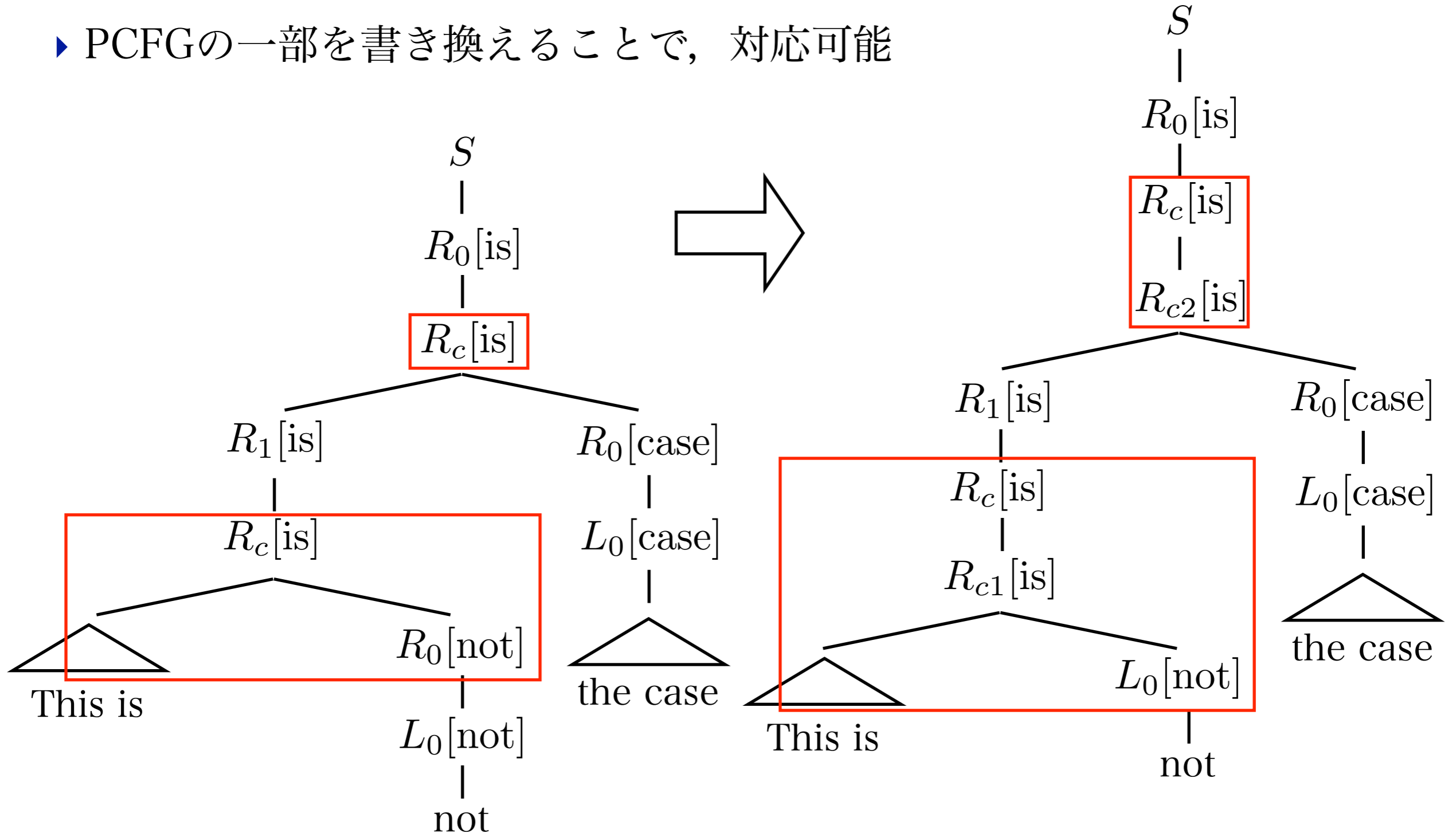
hungryとdogに依存関係があるとき、
hungryはdogの隣に出現しやすいだろう。

* hungry the dog

品詞でも同じことが言える？

Extended Valence Grammar

- ▶ 各headに最も近いargumentと，そうでないargumentに関する分布を変化させる
- ▶ PCFGの一部を書き換えることで，対応可能



Lexicalized EVG

- ▶ これまでのモデルは、全て品詞同士の関係しか見ない
- ▶ 品詞同士でよく表れる関係を捉える
- ▶ Penn Treebankの場合、品詞数は45
- ▶ 自動詞/他動詞 の違いもきちんと捉えられない
- ▶ headの単語を用いて、argumentの品詞を推定

$p(\text{NN}|\text{R}, \text{is}, \text{VBD}, v)p(\text{case}|\text{NN})$

This is not the case
DT VBZ RB DT NN

Head automatonで表現可能
単語の情報は非常にスパース
⇒スムージングを行う

LEVGまでの結果

▶ Penn Treebank

- Section 2-21を訓練データ (長さ10以下)
- Section 23 をテストデータ (長さ10以下)
- Kleinの論文では0-24の全てを使うが、最近では訓練とテストに分ける方が多い

DMV	43.2
EVG	53.3
EVG+smoothed	65
LEVG+smoothed	68.8
shared logistic normal	62.4
TSG+DMV	67.7

Unambiguity Regularization

▶ 一般のEMアルゴリズム

$$\log p(X|\theta) \geq F(q, \theta) = \sum_Y q(Y) \log \frac{p(X, Y)}{q(Y)}$$

$$F(q, \theta) = \sum_Y q(Y) \log(p(X)p(Y|X)) - \sum_Y q(Y) \log q(Y)$$

$$= \log p(X|\theta) - \sum_Y q(Y) \log \frac{q(Y)}{p(Y|X)}$$

$$= \log p(X|\theta) - \text{KL}(q(Y) || p(Y|X))$$

▶ $F(q, \theta)$ を, q と θ に関して交互に最適化

$$\text{E-step: } q^{t+1} = \arg \max_q F(q, \theta) = \arg \min_q \text{KL}(q(Y) || p(Y|X))$$

$$\text{M-step: } \theta^{t+1} = \arg \max_{\theta} F(q^{t+1}, \theta) = \arg \max_{\theta} \log \langle p(X, Y) \rangle_{q^{t+1}}$$

Unambiguity Regularization

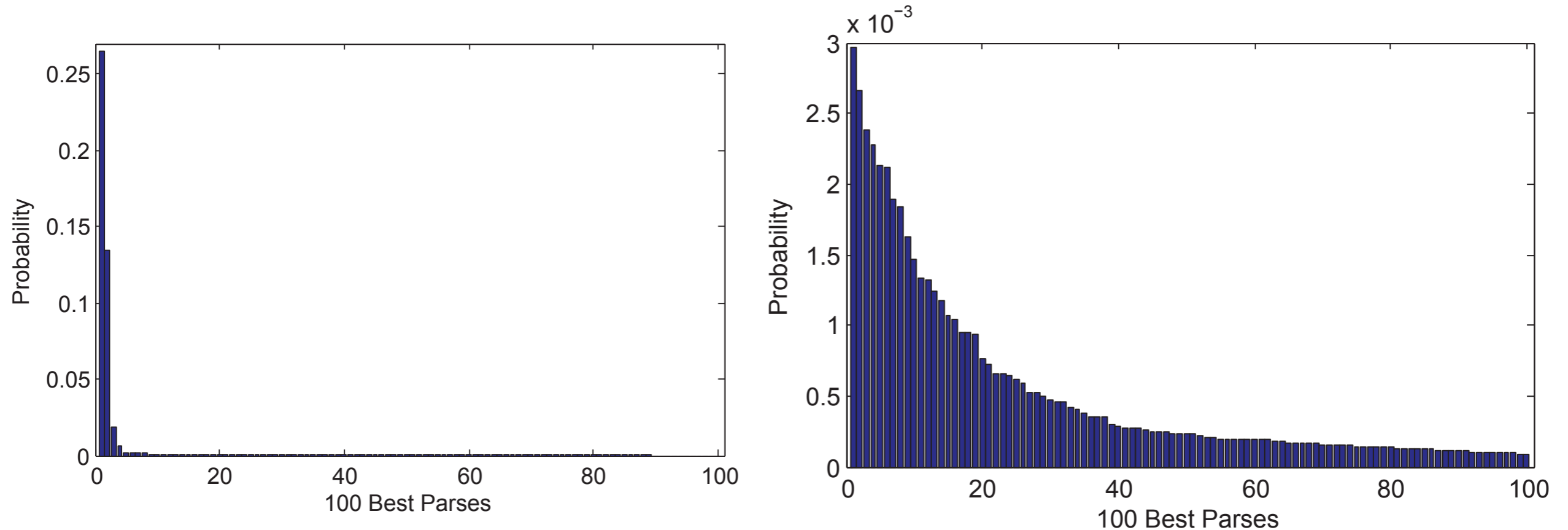
- ▶ 代わりに次の量を最大化する

$$F(q, \theta) = \log(X|\theta) - \left(\text{KL}(q(Y)||p(Y|X)) + \sigma \sum_i H(y_i) \right)$$

$$H(y_i) = - \sum_{y_i} q(y_i) \log q(y_i)$$

- ▶ Posterior regularization と呼ばれる (Gillenwater, et al., 2010)
- ▶ q を求める際, そのエントロピーが小さくなるような制約を入れている

自然言語のUnambiguity



- ▶ 自然言語の文章は曖昧か？ ⇒ 多くの文章は，1つの解釈しか持たない
- ▶ ある文に対して，
 - (左) Berkeley Parser, (右) PCFGのEMによる推定
で，100 Best parses にそれぞれ，どれだけの確率を割り当てたか

Unambiguity Regularization

- ▶ 通常Posterior regularizationは勾配法などで近似する
- ▶ 事後分布のエントロピーを小さくするような仮定を置くと，解析的に解ける
- ▶ 結果
 - 従来のEMアルゴリズムで，Eステップの前にパラメタを $\frac{1}{1-\sigma}$ 乗すれば良いだけ！
 - 従来研究で，Hard-EMの性能が良いことが示されていた
 - Hard-EMは，このモデルで $\sigma = 1$ と置いた場合に相当する
 - 制約の強さを最大にした場合（最も曖昧でないような学習を行う）

結果

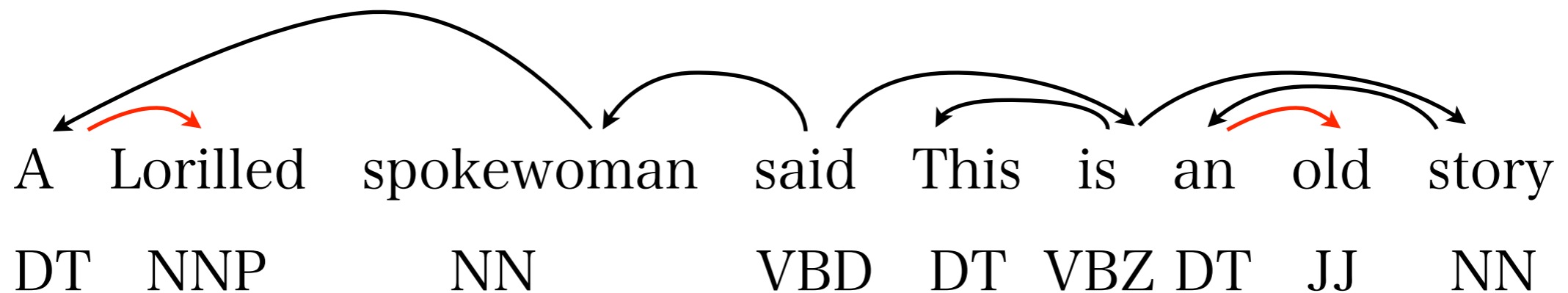
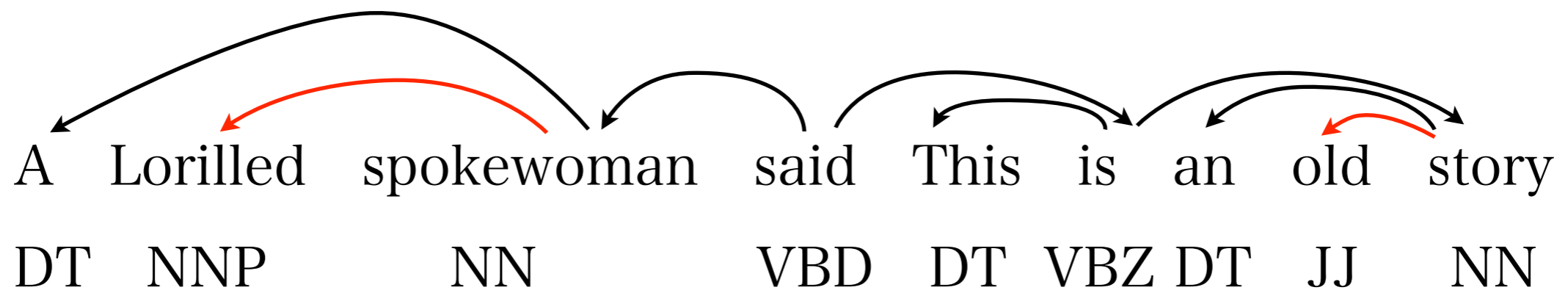
Value of σ	Testing Accuracy		
	≤ 10	≤ 20	All
0 (standard EM)	46.2	39.7	34.9
0.25	53.7	44.7	40.3
0.5	51.9	42.9	38.8
0.75	51.6	43.1	38.8
1 (Viterbi EM)	58.3	45.2	39.4

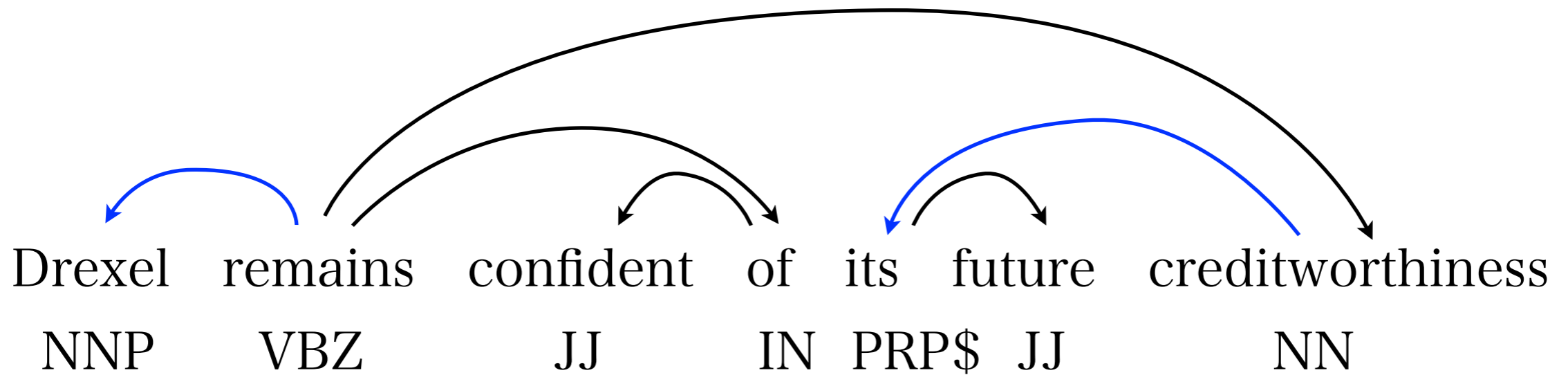
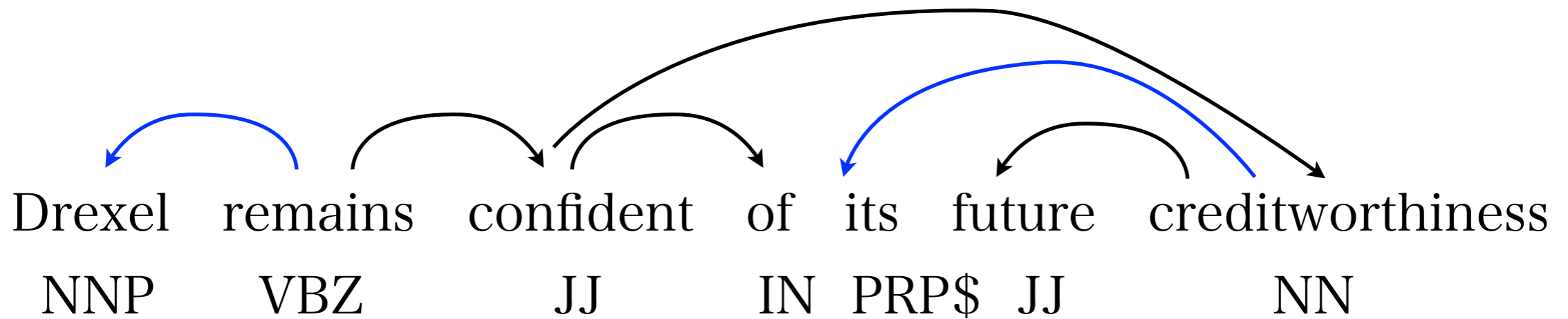
Extended Models			
UR-Annealing on E-DMV(2,2)	71.4	62.4	57.0
UR-Annealing on E-DMV(3,3)	71.2	61.5	56.0
L-EVG (Headden et al., 2009)	68.8	-	-
LexTSG-DMV (Blunsom and Cohn, 2010)	67.7	-	55.7

- ▶ UR-Annealing : σ の値を1から0へ徐々に下げていく
- ▶ 英語で現在の state-of-the-art を達成

DMV + unambiguity での解析結果

- ▶ 簡単なDMV + $\sigma=0.25$ での結果を示す (精度: 57.8)
- ▶ Annealing したところ, 良い結果が得られなかった (バグ?)





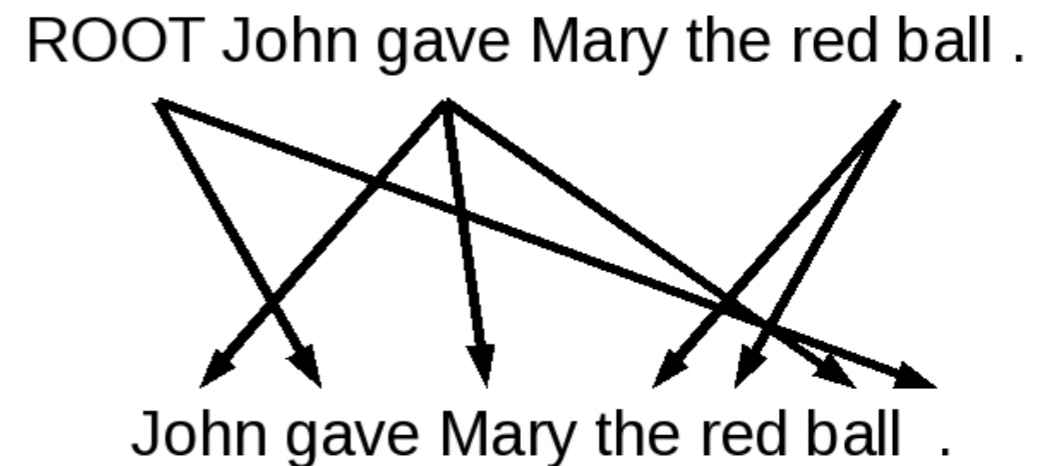
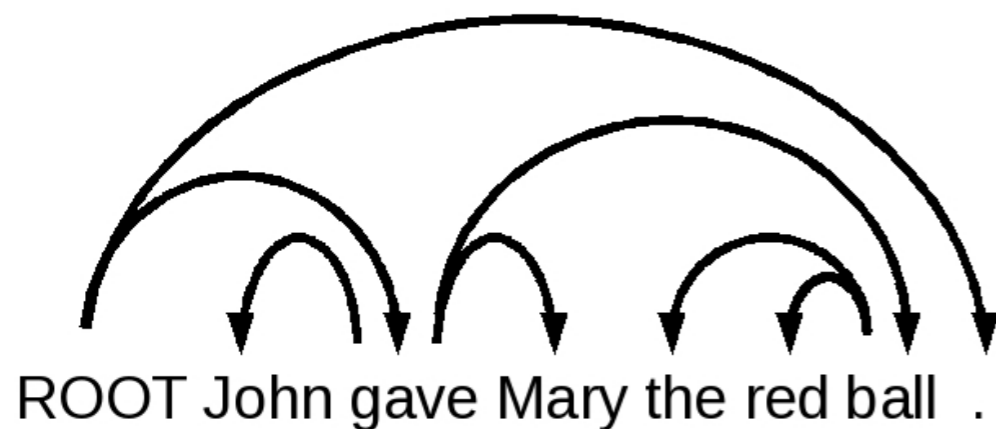
DMVとは異なる方針

- ▶ これまでのDMVの拡張は、全てInside Outsideを用いることの出来るHead automatonモデル
- ▶ 最近、いくつかこれ以外の方針が提案されている
 - (Brody, 2010) : アラインメントのIBMモデルを用いる
 - (Marecek, 2012) : reducibility + Projectiveな木を作るGibbsサンプリング
- ▶ Head automatonにとらわれないモデル化が可能
- ▶ 問題点
 - 推論をどうするか？
 - Projectivityを満たす木を作れるか

IBMモデルを用いた教師なし解析

▶ IBMモデルとDMVとの類似点

- IBMモデル 1 : よく出現する単語の組を対応付ける
- IBMモデル 2 : 単語の出現場所に応じて対応付ける (valence ?)
- IBMモデル 3 : 各単語が結びつく, target側の単語数をモデル化 (valence)



IBMモデルを用いた教師なし解析

Corpus	M 1	M2	M3	R-br
WSJ10	25.42	35.73	39.32	32.85
Dutch10	25.17	32.46	35.28	28.42
Danish10	23.12	25.96	41.94	16.05 *

- ▶ DMVよりは低いが, Right-branchを上回る
- ▶ このモデルは, Projectiveな木を作らないことに注意
- ▶ そのような制約をうまく入れれば, 更に性能が上がる可能性?

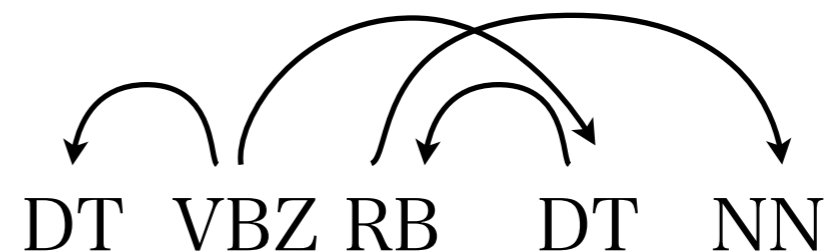
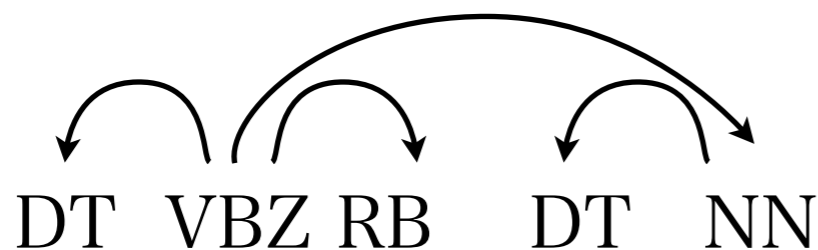
Projectiveな木を作る Gibbs sampling

- ▶ Inside-Outside を用いずに, Projectiveな木を直接推定する
- ▶ Gibbs Samplingで, 局所的な係り関係を書き換える
- ▶ 生成モデルとしてはNon Projectiveな木も作るが, 推定の範囲を Projectiveなモデルに限定している
 - deficientなモデルを定義していることと同等?

基本となるモデル

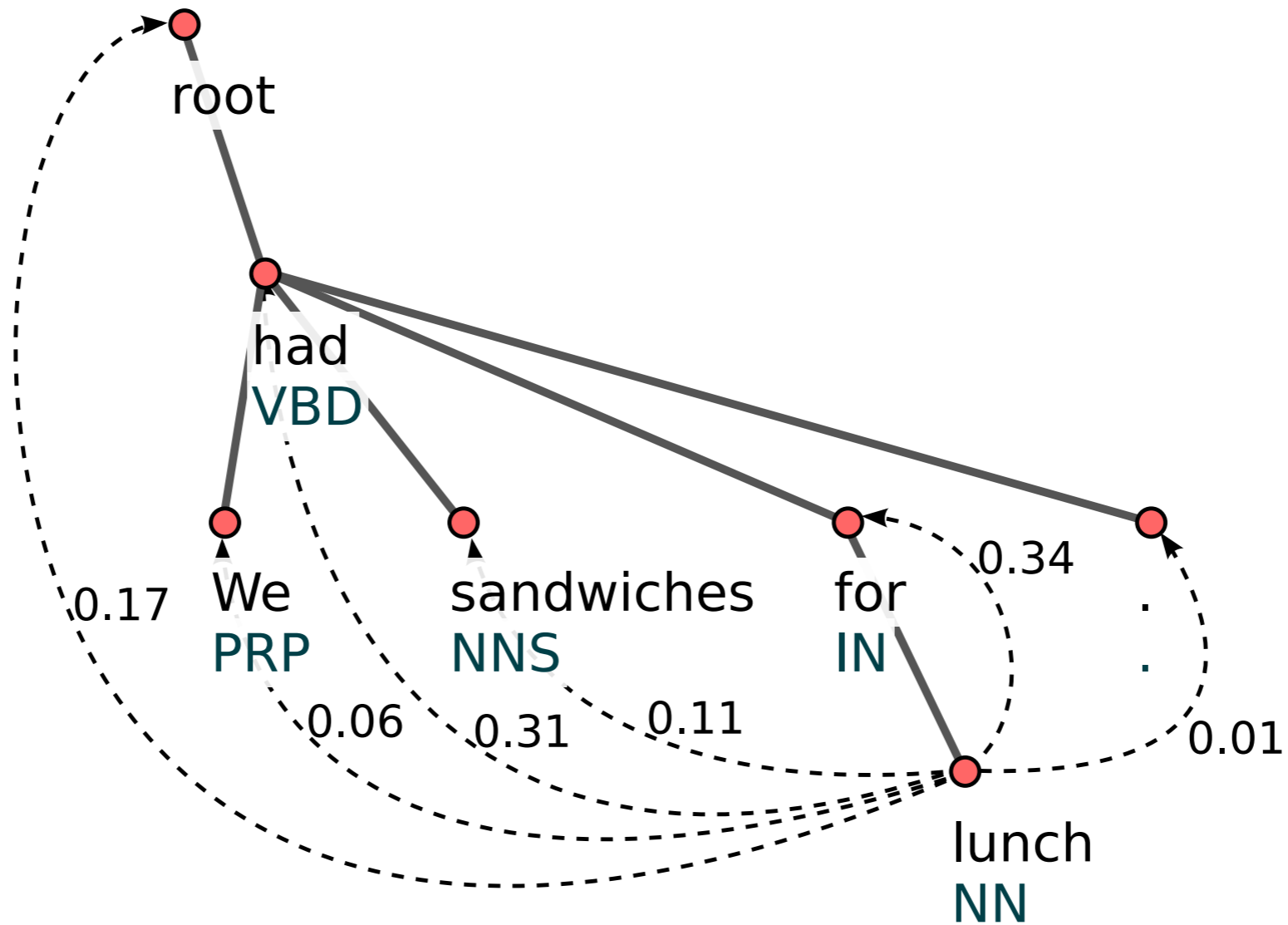
- ▶ edge model
- ▶ 各品詞が取りうるargumentの分布が, Dirichlet分布から生成される
- ▶ よく起こりやすい head /dependent を捉える
- ▶ 必ずしもProjectiveな木を作らない

$$P_{treebank} = \prod_{i=1}^n P_{edge}(t_i | t_{\pi(i)}) = \prod_{i=1}^n \frac{c^{-i}(\text{"}t_i, t_{\pi(i)}\text{"}) + \alpha}{c^{-i}(\text{"}t_{\pi(i)}\text{"}) + \alpha|T|}$$



DMVでは確率を割り当てない

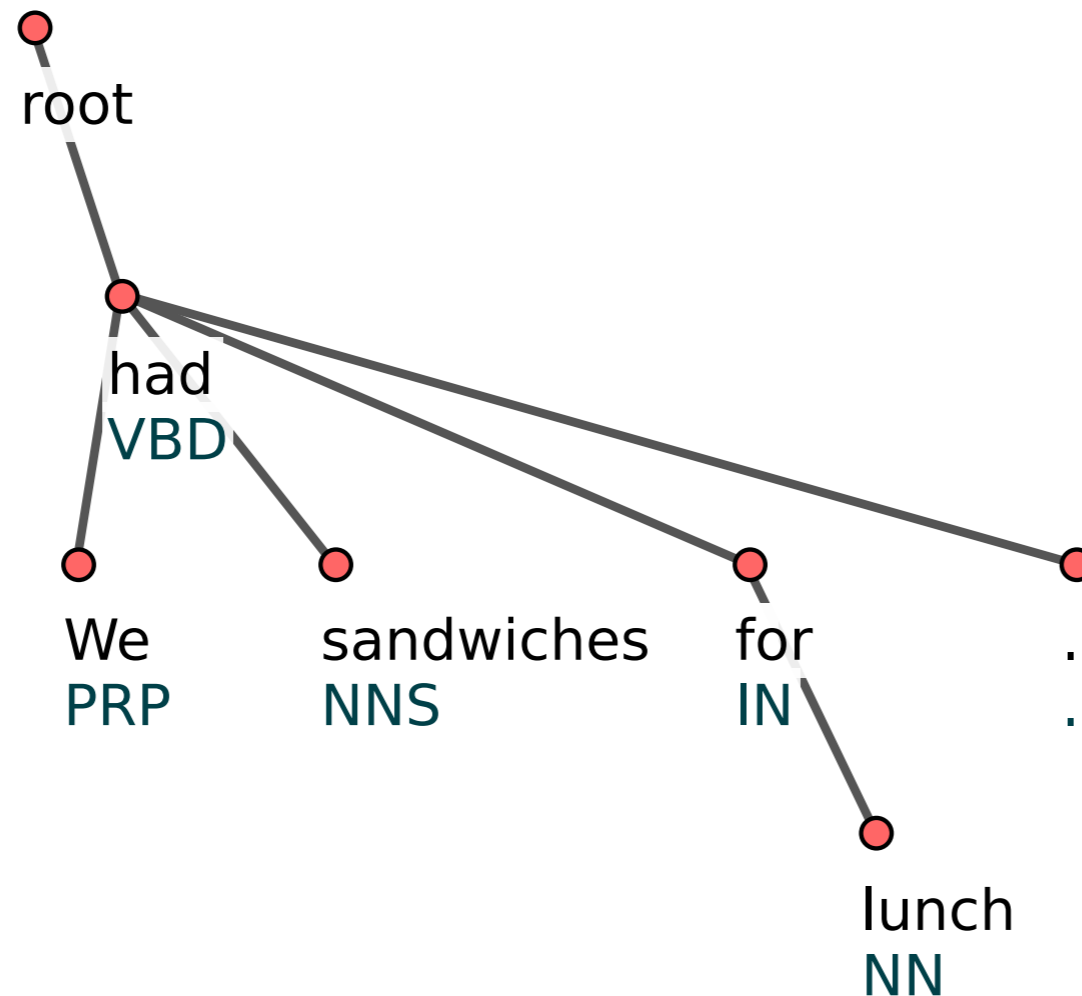
ナイーブな Gibbs sampling



- ▶ 木から lunch を削除する
- ▶ 親を再推定
- ▶ We, sandwiches を選択した場合, projective でなくなる

Projectivityを保つための準備

- ▶ 係り関係をbracketで表現する
- ▶ n個の単語に対して， n個のbracket
- ▶ 各bracketは， その深さの単語を， 1語のみ持つ



((We) had (sandwiches) (for (lunch)) (.))

Gibbs sampling

- ▶ bracketを1つ削除する
- ▶ 新しいbracketを，分布に応じて選択する
- ▶ 1回の変化で，木を大きく書き換えることが出来る

((We) had (sandwiches) (for (lunch)) (.))

((We) had sandwiches (for (lunch)) (.))



(((We) had) sandwiches (for (lunch)) (.))

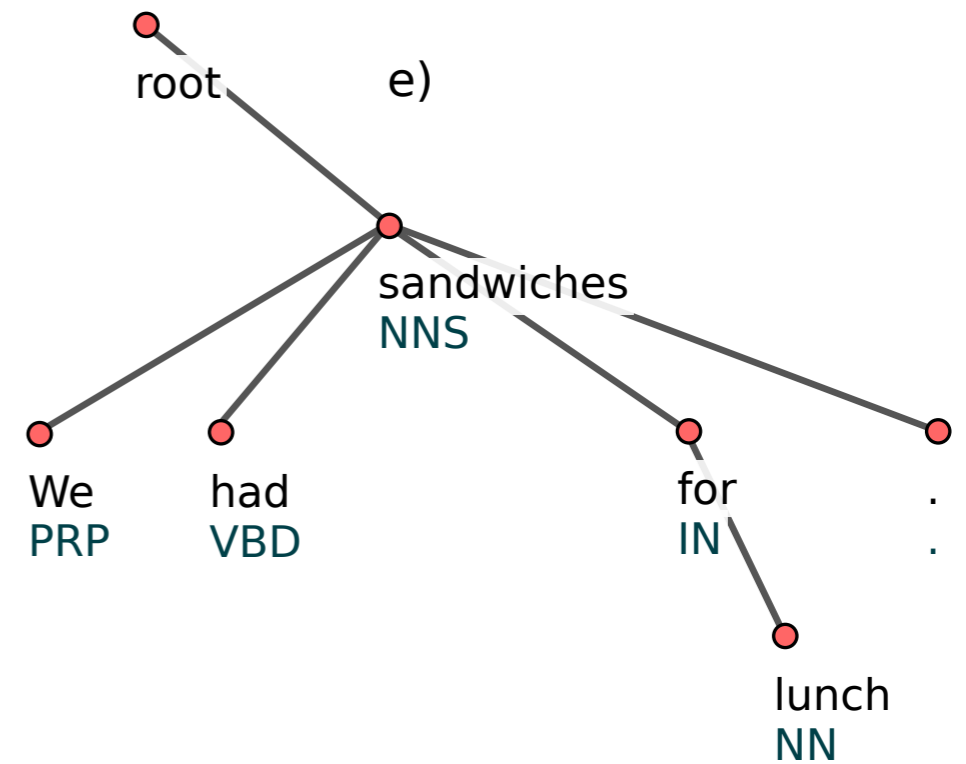
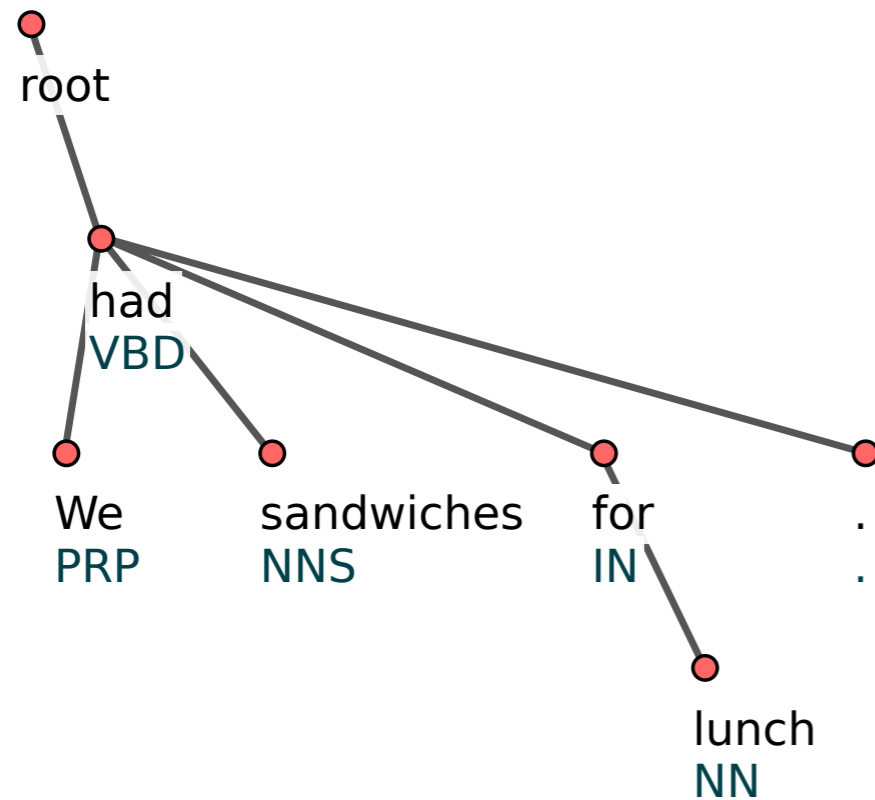
((We) (had) sandwiches (for (lunch)) (.))

((We) had (sandwiches) (for (lunch)) (.))

((We) had (sandwiches (for (lunch))) (.))

((We) had (sandwiches (for (lunch)) (.)))

例えば



((We) (had) sandwiches (for (lunch)) (.))

((We) had (sandwiches) (for (lunch)) (.))

- ▶ EMのlocal moveより効率が良いそう？ ⇒ 初期化の影響はないらしい(!)
- ▶ DMVでも似たような局所的なGibbsは適応できる？

実際のモデル

- ▶ 様々なコンポーネントのProduct of Experts
- ▶ Projectivityを仮定しない代わりに，豊富な情報を組み込むことが可能に

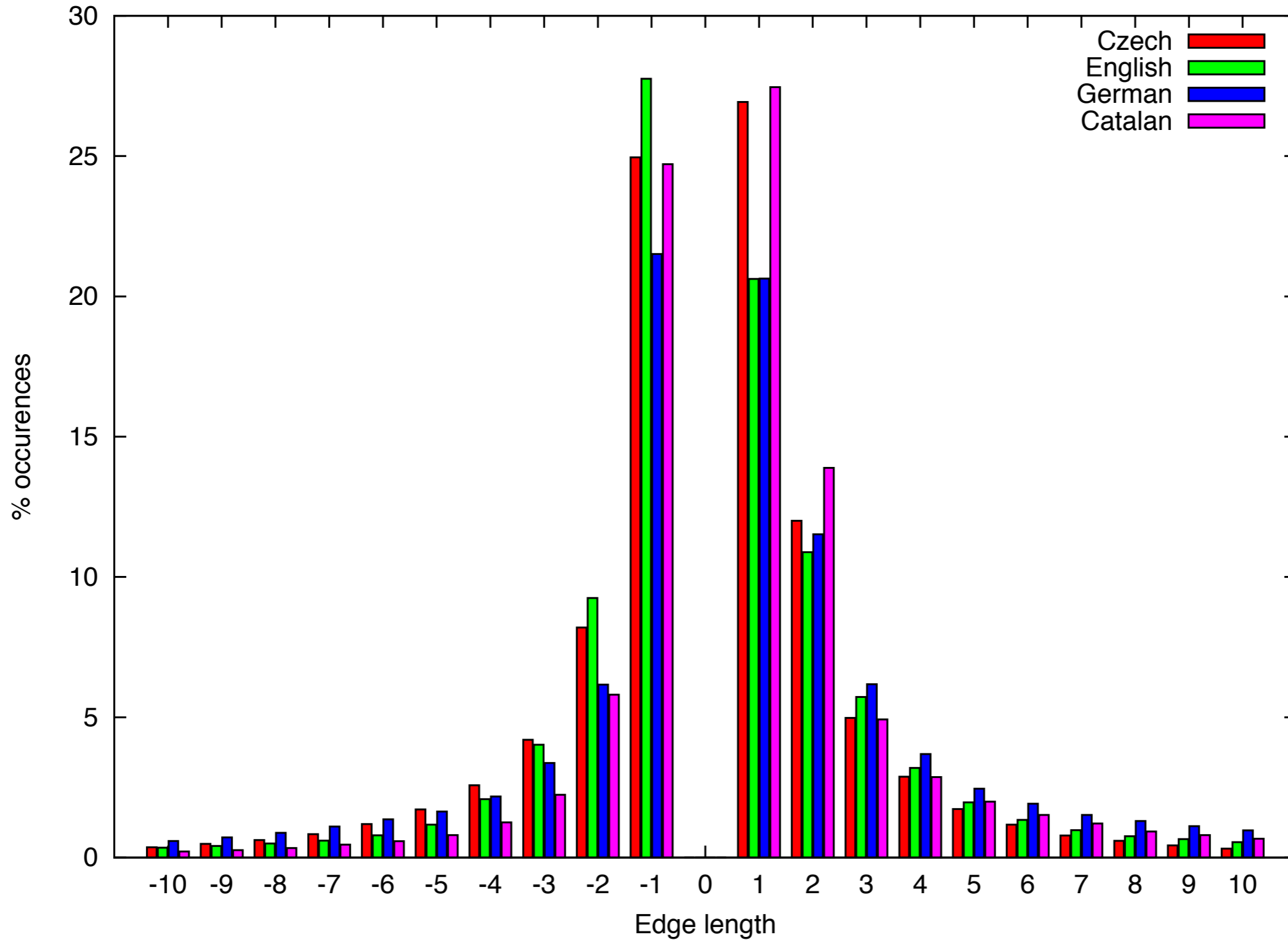
$$\begin{aligned}
 P_{treebank} &= \prod_{i=1}^n P_{etd}(i, \pi(i)) P_{fdx}(f_i, \pi(i)) P_d(i, \pi(i)) P_r(i) = \\
 &= \prod_{i=1}^n \frac{c^{-i}(\text{"}t_i, t_{\pi(i)}, dir(i, \pi(i))\text{"}) + \alpha_{etd}}{c^{-i}(\text{"}t_{\pi(i)}, dir(i, \pi(i))\text{"}) + \alpha_{etd} \cdot |T|} \\
 &\quad \frac{c^{-i}(\text{"}t_i, f_i^L, f_i^R\text{"}) + \frac{\beta_0}{F(w_i)} P_0(f_i^L + f_i^R)}{c^{-i}(\text{"}t_i\text{"}) + \frac{\beta_0}{F(w_i)}} \\
 &\quad \frac{1}{\epsilon_d} \frac{1}{|i - \pi(i)|^\gamma} \quad \boxed{\text{distance model}} \\
 &\quad \frac{1}{\epsilon_r} R(desc(i))^\delta. \quad \boxed{\text{reducibility model}}
 \end{aligned}$$

distance model

$$P_d(d, g) = \frac{1}{\epsilon_d} \left(\frac{1}{|d - g|} \right)^\gamma$$

- ▶ 言語に（恐らく普遍的な）性質
- ▶ 一番近い単語に係りやすい
- ▶ DMVでは，この制約を陽に入れることが難しい
- ▶ valence では，head / dependent の距離そのものは，パラメタ化されていない

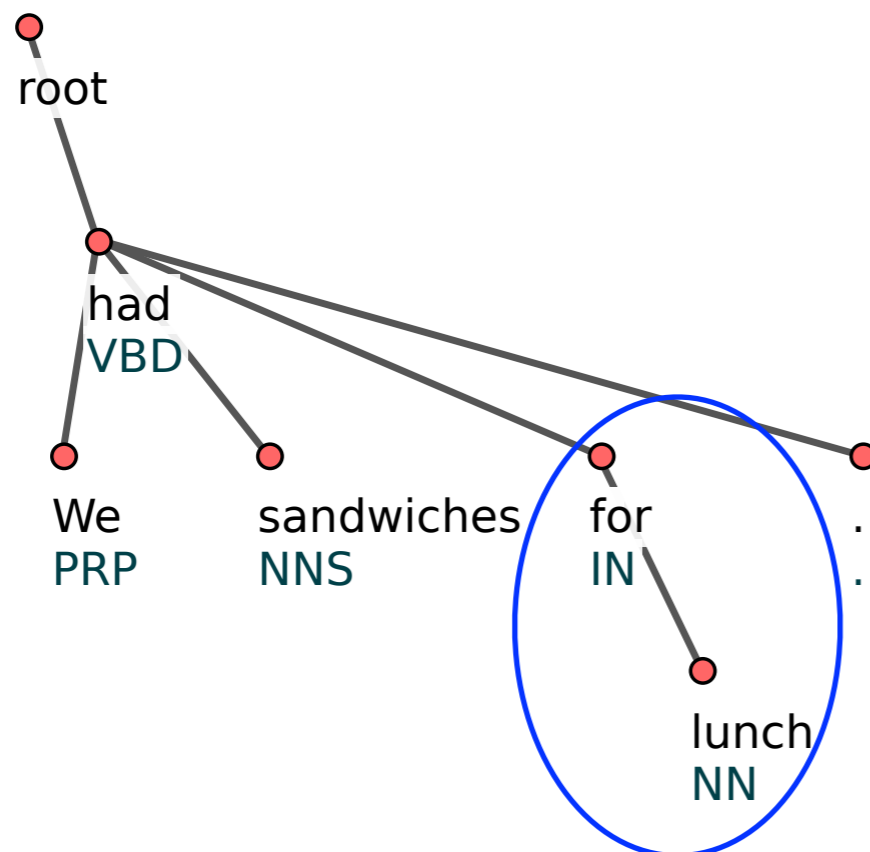
言語毎の距離の統計



reducibility model

$$P_r(i) = \frac{1}{\epsilon_r} R(desc(i))^\delta$$

- ▶ 文の中で、dependentになりやすい部分（例えば前置詞句）は、削除しても意味が通る
- ▶ 品詞nグラムに対するreducibility = 削除しやすさ，を計算する
- ▶ $R(desc(i))$ = 単語 w_i をルートとする部分木のnグラムのreducibility



IN NN というバイグラムは、
reducibilityが高い
⇒ そのような部分木を作る係
り関係にバイアスがかかる

reducibilityの学習

- ▶ 大規模コーパスから学習する
- ▶ ある品詞nグラム = g が削除可能である (reducibilityが高い) とき,
 - コーパス中の g の品詞の列を探す
 - その部分を削除した単語列が, コーパス中に出現するかどうか, を調べる

g = IN NN NN

Motorola	is	fighting	back	against	junk	mail
NNP	VBZ	VBG	RB	IN	NN	NN

reducibilityの学習

- ▶ 大規模コーパスから学習する
- ▶ ある品詞nグラム = g が削除可能である (reducibilityが高い) とき,
 - コーパス中の g の品詞の列を探す
 - その部分を削除した単語列が, コーパス中に出現するかどうか, を調べる

g = IN NN NN

Motorola is fighting back

この文が他の箇所でも出現していれば, 高いスコアがつく

reducibility score

unigrams	reduc.	bigrams	reduc.	trigrams	reduc.
VB	0.04	VBN IN	0.00	IN DT JJ	0.00
TO	0.07	IN DT	0.02	JJ NN IN	0.00
IN	0.11	NN IN	0.04	NN IN NNP	0.00
VBD	0.12	NNS IN	0.05	VBN IN DT	0.00
CC	0.13	JJ NNS	0.07	JJ NN .	0.00
VBZ	0.16	NN .	0.08	DT JJ NN	0.04
NN	0.22	DT NNP	0.09	DT NNP NNP	0.05
VBN	0.24	DT NN	0.09	NNS IN DT	0.14
.	0.32	NN ,	0.11	NNP NNP .	0.15
NNS	0.38	DT JJ	0.13	NN IN DT	0.23
DT	0.43	JJ NN	0.14	NNP NNP ,	0.46
NNP	0.78	NNP .	0.15	IN DT NNP	0.55
JJ	0.84	NN NN	0.22	DT NN IN	0.59
RB	2.07	IN NN	0.67	NNP NNP NNP	0.64
,	3.77	NNP NNP	0.76	IN DT NN	0.80
CD	55.6	IN NNP	1.81	IN NNP NNP	4.27

実験結果

- ▶ 多くの言語において、既存のモデルを上回る
- ▶ 英語では unambiguity regularization のほうが性能が上

CoNLL			≤ 10 tokens		all sentences	
language	code	year	Gillen.2011	our parser	Spitkov.2011	our parser
Arabic	ar	06	–	40.5	16.6	26.5
Arabic	ar	07	–	42.4	49.5	27.7
Basque	eu	07	–	32.8	24.0	27.2
Bulgarian	bg	06	58.3	59.0	43.9	49.0
Catalan	ca	07	–	63.5	59.8	47.0
Czech	cs	06	53.2	58.9	27.7	49.5
Czech	cs	07	–	67.6	28.4	50.7
Danish	da	06	45.9	52.8	38.3	40.4
Dutch	nl	06	33.5	42.4	27.8	41.7
English	en	07	–	64.1	45.2	49.2
German	de	06	46.7	60.8	30.4	44.8
Greek	el	07	–	35.8	13.2	25.4
Hungarian	hu	07	–	63.2	34.7	51.1
Italian	it	07	–	50.5	52.3	43.3
Japanese	ja	06	57.7	68.6	50.2	52.5
Portuguese	pt	06	54.0	66.0	36.7	54.9
Slovenian	sl	06	50.9	51.0	32.2	37.8
Spanish	es	06	57.9	67.3	50.6	51.9
Swedish	sv	06	45.0	62.9	50.0	49.9
Turkish	tr	07	–	18.6	35.9	20.9
<i>Average:</i>			50.3*	59.0*	37.4	42.1

教師なし係り受け解析まとめ

▶ 2つの方向性

- PCFGに変換した上で、様々な情報を加えていく
 - Extended Model, TSG + DMV など
- (Projectiveな) 木を生成する, 新しい生成モデルを定義する
 - Head automatonにとらわれずに柔軟な分布を設定できる
 - 今後はこちらが流行るかも？

▶ 2012年の状況

- 生のコーパスのみからの推定は, 2009年以降あまり進歩していない
- DMVからモデルを変えずに7割程度の精度を達成 (今年) したことを考えると, 2004年からもあまり進歩していないような気も…
- 最近は他の方向性 (多言語のデータを同時に学習, 英語でのパラメタを他の言語に投射する) が流行っている (Naseem, et al., 2012; McDonald, et al., 2011)
 - しかし言語の本質を考えるとという点では, 問題から逃げている？
 - 工学的にはこちらも重要

参考文献

- ▶ Abney, S. P. 1987. *The English Noun Phrase in its Sentential Aspect*. PhD thesis, MIT.
- ▶ Alshawi, H. 1996 Head automata and bilingual tiling: Translation with minimal representations. *In ACL*
- ▶ Blunsom, P., and Cohn, T. 2010. Unsupervised induction of tree substitution grammars for dependency parsing. *In EMNLP*
- ▶ Brody, S. 2010. It depends on the translation: Unsupervised dependency parsing via word alignment. *In EMNLP*
- ▶ Carroll, G. and Charniak, E. 1992. Two experiments on learning probabilistic dependency grammars from corpora. *In Working Notes of the Workshop Statistically-Based NLP Techniques*
- ▶ Cohen, S.B., Gimpel, K., and Smith, N.A. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. *In NIPS*
- ▶ Cohen, S.B., and Smith, N.A. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. *In NAACL-HLT*
- ▶ Collins, M. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, The University of Pennsylvania.
- ▶ Gillenwater, J., Ganchev, K., Graca, J., Pereira, F., and Taskar, B. 2011. Posterior sparsity in unsupervised dependency parsing. *In JMLR*
- ▶ Klein, D. and Manning, C.D. 2002. A generative constituent-context model for improved grammar induction. *In ACL*
- ▶ Klein, D. and Manning, C.D. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. *In ACL*
- ▶ Klein, D. 2005. *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University
- ▶ Marecek, D. 2012. *Unsupervised Dependency Parsing*. Ph.D. thesis, Unsupervised Dependency Parsing
- ▶ McDonald, R.T., Petrov, S., and Hall, K., Multi-source transfer of delexicalized dependency parsers, *In EMNLP*
- ▶ Naseem, T., Barzilay, R., and Globerson, A., Selective Sharing for Multilingual Dependency Parsing, *In ACL*
- ▶ Paskin, M.A. 2001. Grammatical bigrams. *In NIPS*
- ▶ Seginer, Y. 2007. *Learning syntactic structure*. Ph.D. thesis, Universiteit van Amsterdam
- ▶ Smith, N.A. 2006. *Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text*. Ph.D. thesis, Department of Computer Science, Johns Hopkins University
- ▶ Spitzkovsky, V.I., Alshawi, H., Jurafsky, D., and Manning, C.D. 2010. Viterbi training improves unsupervised dependency parsing. *In CoNLL*
- ▶ Tu, K and Honavar, V. 2012. Unambiguity regularization for unsupervised learning of probabilistic grammars. *In EMNLP*
- ▶ William P. Headden III, W.P., Johnson, M. and McClosky, D. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. *In NAACL-HLT*