WEB-SCALE FEATURES FOR FULL-SCALE PARSING

Mohit Bansal and Dan Klein, ACL 2011, pp. 693-702
読み手: 岡崎　直観

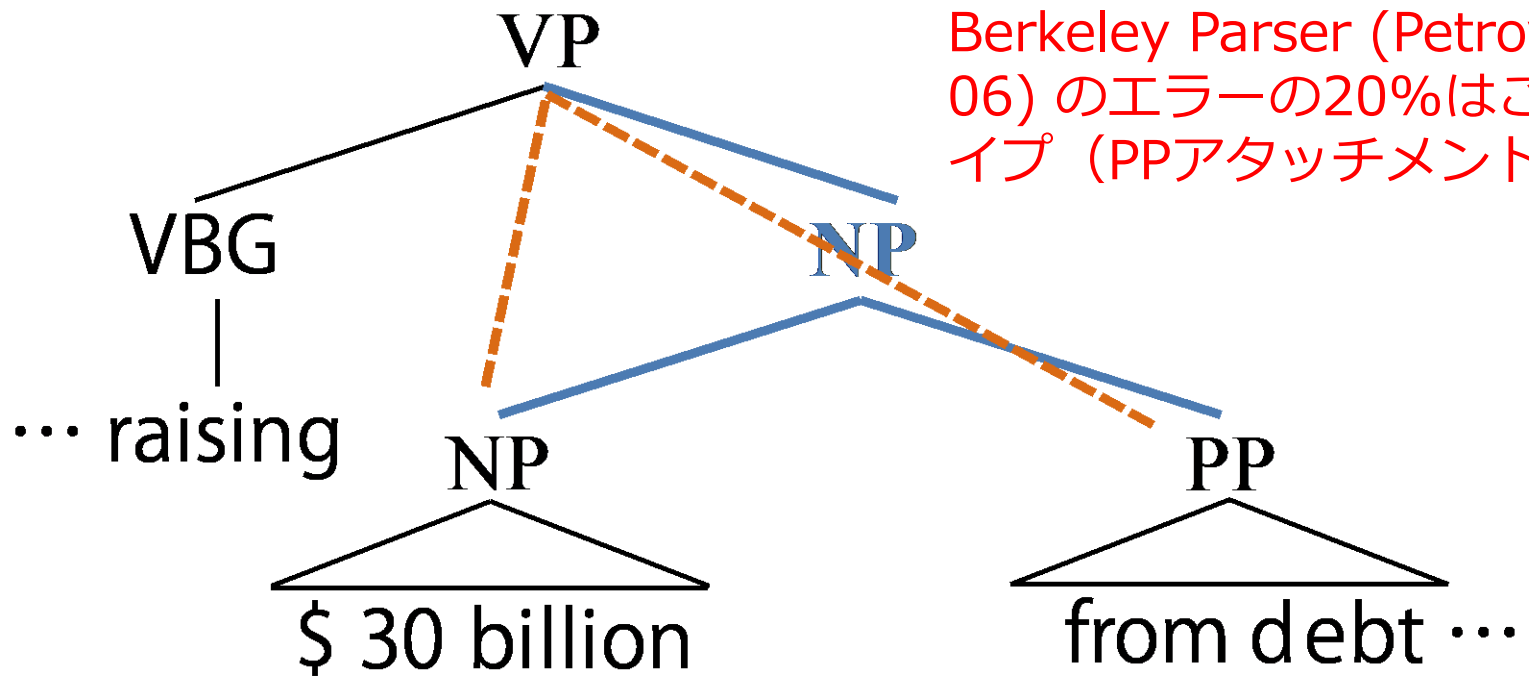
本論文について

- ラベル無しデータ（Web全体の統計量）を使い，常識のような知識を獲得し，言語処理のタスクを改善する例
  - 一見するとパーザの細かい話に見えてしまうが，述語項構造解析や固有表現抽出などのタスクにも使えそう
- 訓練データから文脈をマイニングするのも興味深い

研究の背景

- 最近のパーザはF1で90を達成（on Penn Treebank）
- アタッチメントなどの構造的なエラーが残されている
  - Berkeley parser (Petrov+ 06) のエラーの20%はPPアタッチメント（次スライド）
  - 他にも色々なタイプのエラーがある（次次スライド）
- このようなエラーを100万語くらいの学習コーパスだけで改善するのは難しい

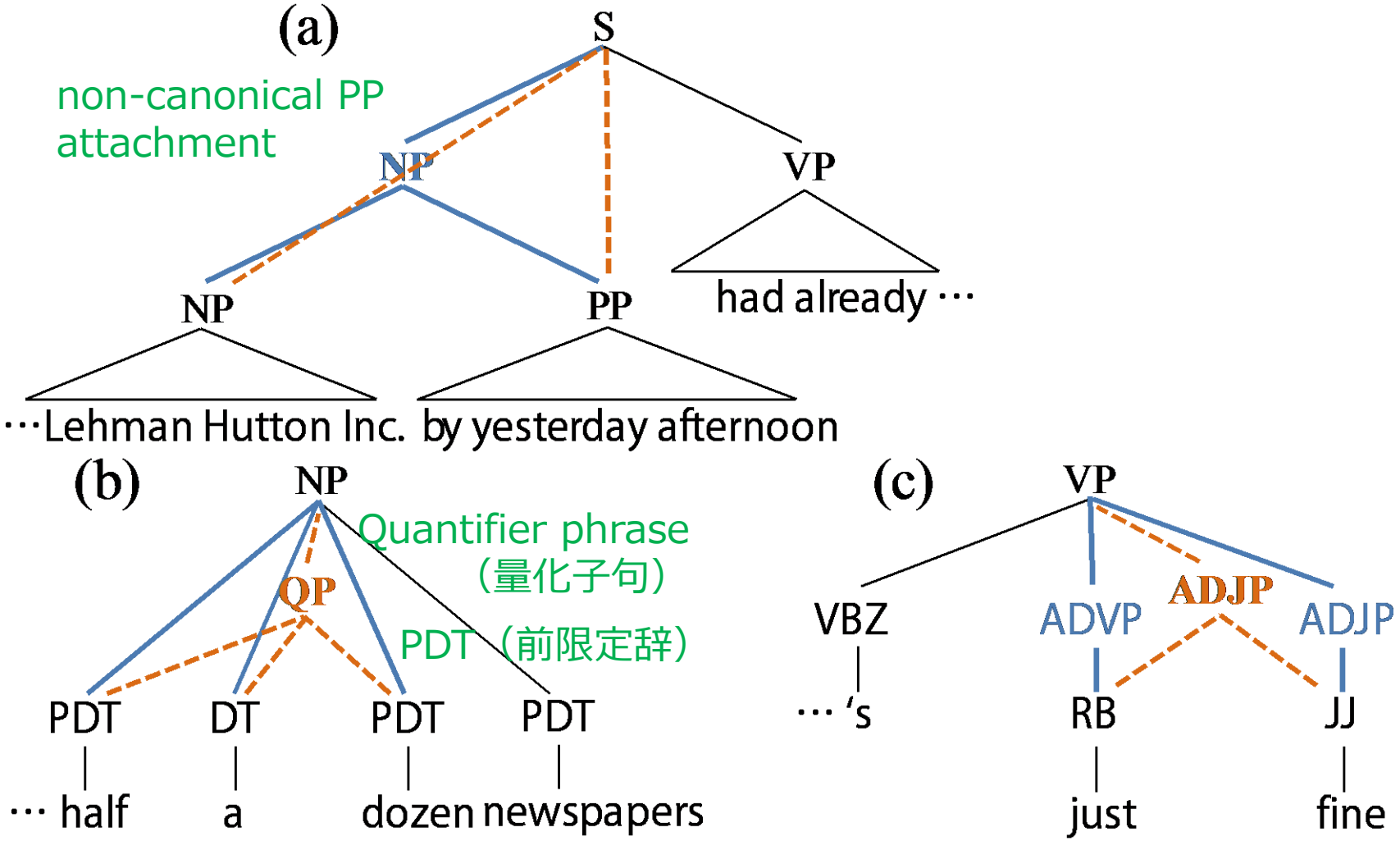
典型的な解析誤り



Berkeley Parser (Petrov+ 06) のエラーの20%はこのタイプ（PPアタッチメント）

The complex financing plan in the S&L bailout law includes raising $ 30 billion from debt issued by the newly created RTC.

その他の解析誤り


(a)
non-canonical PP attachment
S
NP
VP
NP
PP
had already …
…Lehman Hutton Inc.
by yesterday afternoon
(b)
NP
Quantifier phrase
(量化子句)
QP
PDT (前限定辞)
PDT
DT
PDT
PDT
… half
a
dozen
newspapers
(c)
VP
VBZ
ADVP
ADJP
ADJP
… ‘s
RB
JJ
just
fine

アタッチメントエラーに対する従来手法

- 親和力 (Lauer 95, Volk 01, Lapata+ 04)
  - 「$x billion from」よりも「raising from」がよく用いられる
  - 「hydrogen ion exchange」において「hydrogen exchange」よりも「hydrogen ion」の方がよく用いられる
- 言い換え表現を調べる (Nakov+ 05, Pitler+ 10)
  - 「raising it from」が用いられるなら「raising」 →「from」
  - 「be $x billion from」が用いられるなら「billion」→「from」
  - 「The $x billion from」が用いられるなら「billion」→「from」

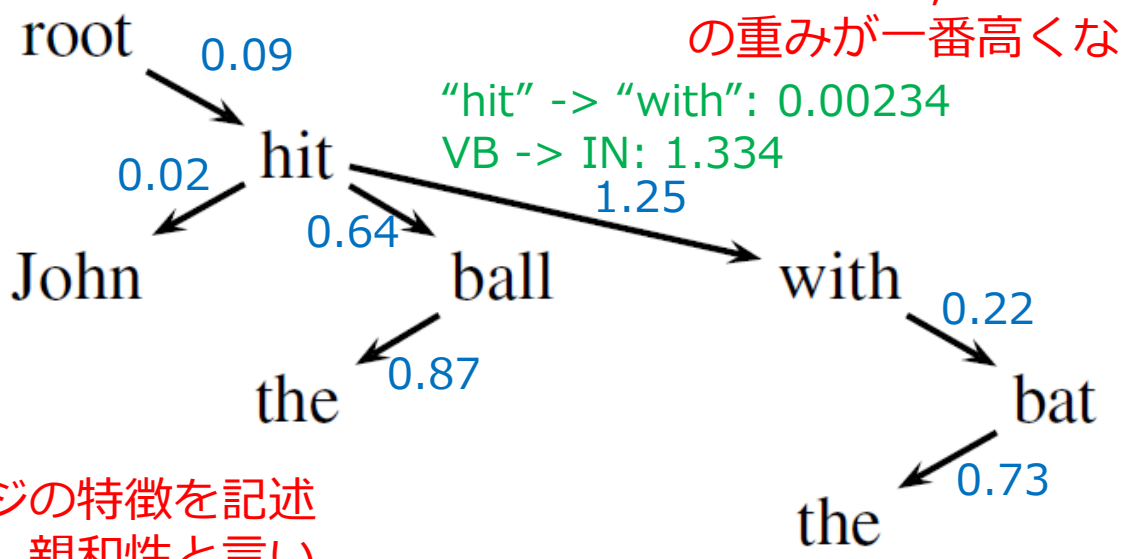
本研究の特色

- 任意のhead-argumentに対応しうる一般的な素性設計
  - 親和力を用いた素性は簡単，言い換え素性は自明ではない
- 既存のパーザに組み込める実用的な素性設計
  - 係り受け解析器としてMST parser (McDonald+ 06)
  - 句構造解析器としてBerkeley parser (Petrov+ 06)
- Google n-gramコーパスの利用
  - 従来はWebページの検索ヒット件数を用いるものが多かった

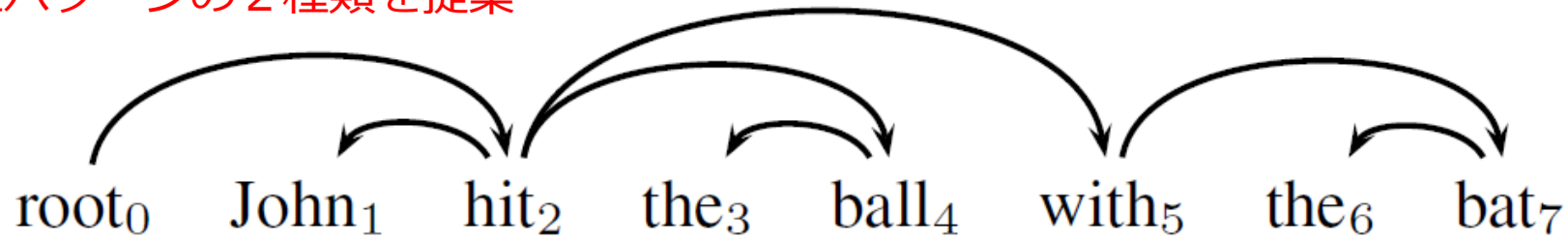
MSTParserの予備知識

- 構文解析: t = argmax $\boldsymbol{w} \cdot \Phi(\boldsymbol{x})$

基本的には，各エッジに素性関数を導入し，グラフ全体で素性の重みが一番高くなる木を選ぶ



本研究は，エッジの特徴を記述する素性として，親和性と言い換えパターンの２種類を提案

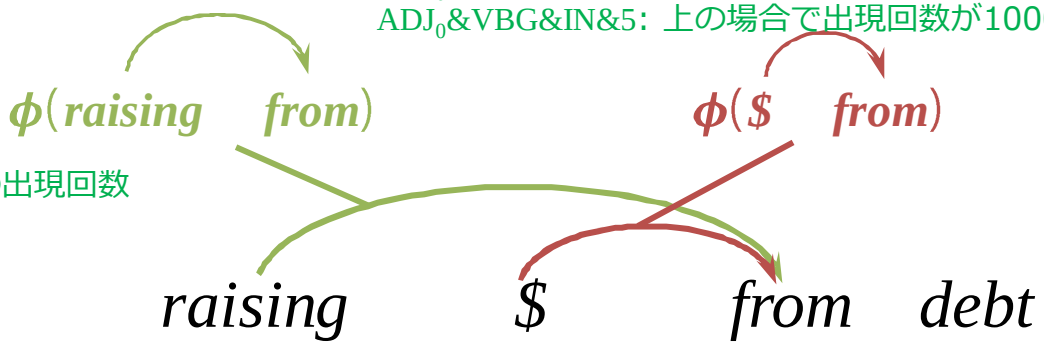


親和性(=隣接性)素性

- 2つの単語hとaが近くにどのくらい出現するか？
  - $\Phi(h, a)$: hとaの係りやすさ
    - 文中でhの後にaが出現する場合: $q \in \{ha, h*a, h**a, h***a\}$
    - 文中でhの前にaが出現する場合: $q \in \{ah, a*h, a**h, a***h\}$
  - すべてのqに対して
    - ADJ: qの出現回数
    - ADJ & POS(h) & POS(a): POS(h)のhとPOS(a)のaが隣接する回数
    - ADJ & POS(x) & POS(y) & b: 上の素性の頻度を離散的にしたもの
      - b = floor($\log_r$(count)/5)*5

ADJ_0: “raising from”の出現回数
ADJ_0&VBG&IN: “raising”がVBGでfromがINの時の“raising from”の出現回数
ADJ_0&VBG&IN&5: 上の場合で出現回数が1000の場合

ADJ_1: “raising * from”の出現回数
…



言い換えパターンのマイニング

- 従来手法ではテストに用いるパターンを手動で作成
  - 「VB it IN」が用いられるなら「VB」→「IN」
  - 「be NN IN」が用いられるなら「NN」→「IN」
  - 「The NN IN」が用いられるなら「NN」→「IN」
- 本研究ではパターンを自動生成する
  - 訓練データから正しい係りを収集: 例 (h, a) = (raising, from)
  - 次のパターンで周辺のコンテキスト*c*を探す: *cha*, *hca*, *hac*
    - 実際には*h*と*a*の品詞毎にマイニングをするらしい
  - 例えば”raising *c* from”で頻出パターンをマイニングすると，*c* = “it”が見つかる

言い換え素性

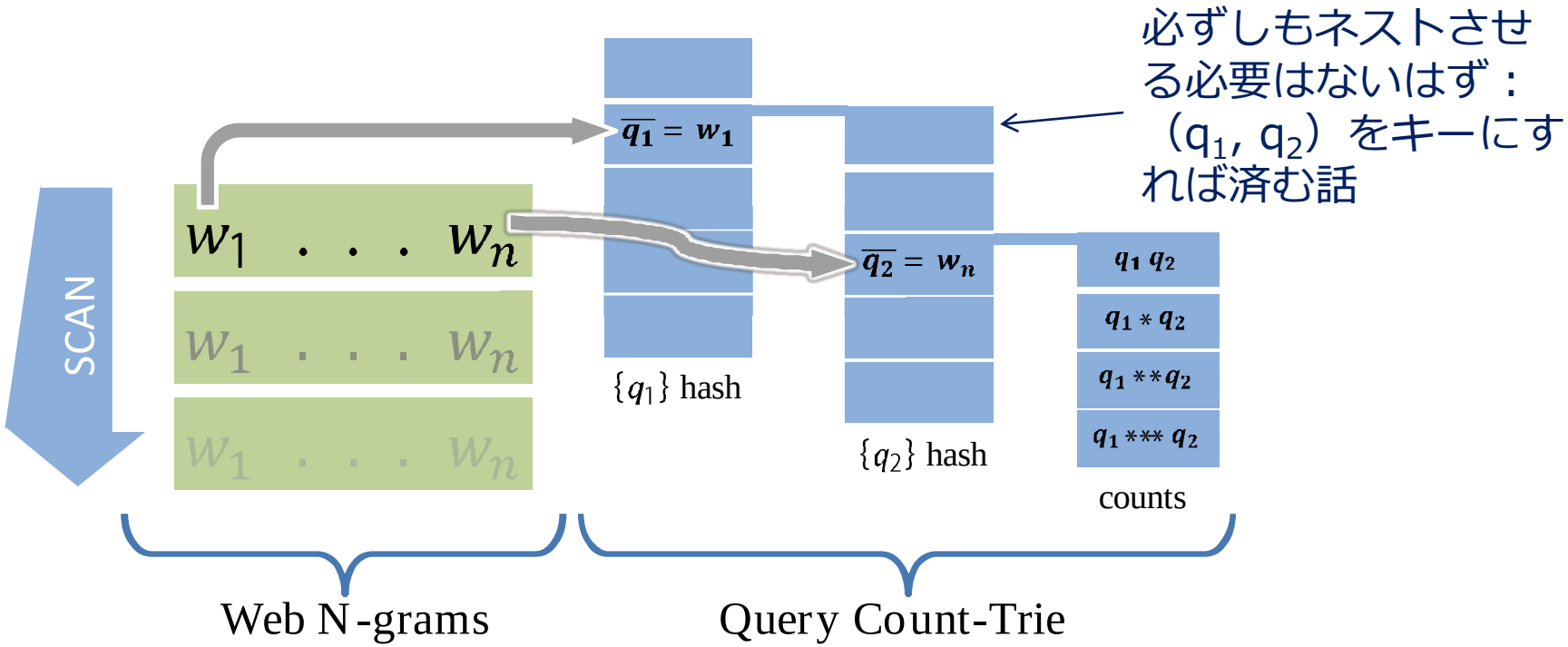
- 単語hとaが特定のパターンでどのくらい出現するか？
  - コンテキストの位置を次のように表現
    - p = “before” if cha; p = “middle” if hca; p = “after” if hac
  - 次のようなテンプレートの素性を定義
    - PARA & POS(h) & POS(a) & c & p & dir
    - PARA & POS(h) & POS(a) & POS(c) & p & dir

hとaの出現順序: {→, ←}

スパースになるかもしれないので

  - 例えば，hcaでc = “it”の場合
    - PARA & POS(h) & POS(a) & c & p & dir
      - PARA&VBG&IN&it&middle&→: 動名詞に続けてit, 前置詞が出現する回数
        - 「raising it from」や「lowering it with」などの表現で素性が共有される
    - PARA & POS(h) & POS(a) & POS(c) & p & dir: PARA&VBG
      - PARA&VBG&IN&PRP&middle&→:動名詞に続けて代名詞, 前置詞が出現する回数
        - 「raising it from」や「providing him with」などの表現で素性が共有される

データ構造



- 本研究の実験では，Google N-gramコーパスに450万件のクエリを発行したが，検索にかかったトータルの時間は115分だった

係り受け解析の実験

- MSTParserに提案した素性を追加
- Penn Treebankで実験
  - 訓練データ: §2-21, 開発データ: §22, 評価データ: §23
  - 品詞タグ付けにはMXPOST (Ratnaparkhi 96) を用いた
  - 素性の次数は１次
- 評価結果
  - 提案手法を用いない２次の素性の場合と比べても改善

	Order 2	+ Web features	% Error Redn.
Dev (sec 22)	92.1	**92.7**	7.6%
Test (sec 23)	91.4	**92.0**	7.0%

句構造解析の実験

- Berkeley parserのtop-k解析木をリランキング
  - リランキングの学習器として平均化パーセプトロンを用いた
  - 素性
    - Berkeley parserが出力した確率値: $\log p(t|w)$
    - 提案手法の素性（Web features）
    - 非局所情報の素性（Configurational features）
  - Top-kをとることの可能性（oracleスコア）

	$k=1$	$k=2$	$k=10$	$k=25$	$k=50$	$k=100$
Dev	90.6	92.3	95.1	95.8	96.2	96.5
Test	90.2	91.8	94.7	95.6	96.1	96.4

← Top-100の中から正しい木を選べればウハウハ

  - 評価結果

Parsing Model	Dev (sec 22) F1	Dev (sec 22) EX	Test (sec 23) F1	Test (sec 23) EX
Baseline (1-best)	90.6	39.4	90.2	37.3
$\log p(t\|w)$	90.4	38.9	89.9	37.3
+ Web features	91.6	42.5	91.1	40.6
+ Configurational features	91.8	43.8	91.1	40.6
+ Web + Configurational	**92.1**	**44.0**	**91.4**	**41.4**

非局所情報と同程度の貢献

非局所情報と提案手法を組み合わせると，さらに改善

解析誤りの減少

- PPアタッチメント（IN）で改善の効果を確認
- その他のタイプのアタッチメントでも改善が確認できる

Arg Tag	# Attach	Baseline	This Work	% ER
NN	5725	5387	5429	12.4
NNP	4043	3780	3804	9.1
IN	4026	3416	3490	12.1
DT	3511	3424	3429	5.8
NNS	2504	2319	2348	15.7
JJ	2472	2310	2329	11.7
CD	1845	1739	1738	-0.9
VBD	1705	1571	1580	6.7
RB	1308	1097	1100	1.4
CC	1000	855	854	-0.7
VB	983	940	945	11.6
TO	868	761	776	14.0
VBN	850	776	786	13.5
VBZ	705	633	629	-5.6
PRP	612	603	606	33.3

Baselineは２次のMSTParser

重用された親和性素性の例

POS_{head}	POS_{arg}	Example (*head, arg*)
RB	IN	*back* → *into*
NN	IN	*review* → *of*
NN	DT	*The* ← *rate*
NNP	IN	*Regulation* → *of*
VB	NN	*limit* → *access*
VBD	NN	*government* ← *cleared*
NNP	NNP	*Dean* ← *Inc*
NN	TO	*ability* → *to*
JJ	IN	*active* → *for*
NNS	TO	*reasons* → *to*
IN	NN	*under* → *pressure*
NNS	IN	*reports* → *on*
NN	NNP	*Warner* ← *studio*
NNS	JJ	*few* ← *plants*

名詞が前置詞に係るかどうかは，hとaの親和性から判断しやすい

動詞が名詞に係る（目的語をとる）かどうかは，hとaが隣接しやすいかどうかを考慮するとよい

重用された言い換え素性の例

POS_{head}	mid-word	POS_{arg}	Example (*head, arg*)
VBN	*this*	IN	*leaned, from*
VB	*this*	IN	*publish, in*
VBG	*him*	IN	*using, as*
VBG	*them*	IN	*joining, in*
VBD	*directly*	IN	*converted, into*
VBD	*held*	IN	*was, in*
VBN	*jointly*	IN	*offered, by*
VBZ	*it*	IN	*passes, in*
VBG	*only*	IN	*consisting, of*
VBN	*primarily*	IN	*developed, for*
VB	*us*	IN	*exempt, from*
VBG	*this*	IN	*using, as*
VBD	*more*	IN	*looked, like*
VB	*here*	IN	*stay, for*
VBN	*themselves*	IN	*launched, into*
VBG	*down*	IN	*lying, on*

bfr-word	POS_{head}	POS_{arg}	Example (*head, arg*)
second	NN	IN	*season, in*
The	NN	IN	*role, of*
strong	NN	IN	*background, in*
our	NNS	IN	*representatives, in*
any	NNS	IN	*rights, against*
A	NN	IN	*review, of*
:	NNS	IN	*Results, in*
three	NNS	IN	*years, in*
In	NN	IN	*return, for*
no	NN	IN	*argument, about*
current	NN	IN	*head, of*
no	NNS	IN	*plans, for*
public	NN	IN	*appearance, at*
from	NNS	IN	*sales, of*
net	NN	IN	*revenue, of*
,	NNS	IN	*names, of*
you	NN	IN	*leave, in*
have	NN	IN	*time, for*
some	NN	IN	*money, for*
annual	NNS	IN	*reports, on*

結論

- ラベル付けされていないコーパスの統計量を，教師あり学習で構築される構文解析の枠組みに取り込めた
- 構文解析に効果的なパターンを自動的にマイニングすることができた
- 従来の機械学習アプローチでは解決することが難しかった解析誤りを改善できた