Max-marginal フィルタを利用した品詞タグ付けと句構造解析の同時学習

†♣ 上垣外 英剛　††♢ 林 克彦　††♢ 平尾 努　†♠ 高村 大也　†♠ 奥村 学　††♢ 永田 昌明
† 東京工業大学
†† 日本電信電話株式会社　NTT コミュニケーション科学基礎研究所
♣kamigaito@lr.pi.titech.ac.jp
♢{hayashi.katsuhiko,hirao.tsutomu,nagata.masaaki}@lab.ntt.co.jp
♠{takamura,oku}@pi.titech.ac.jp

1 はじめに

句構造解析で一般的に用いられる X-bar 文法では、木構造により、遠く離れた単語間の関係を考慮出来る長所があるが、解析途中で各単語に与えられた品詞を記憶し、用いる事は、計算量の観点から難しい。また品詞タグ付けでは、隣接する単語の品詞を記憶し、解析に用いる事が可能であるが、遠く離れた単語間の関係を考慮する事も計算量の観点から難しい。句構造解析と品詞タグ付けを同時に行う事が出来れば、両者の長所を利用しながら、欠点を補う事となり、解析精度の向上が期待出来る。文献 [9] では双対分解により句構造解析と品詞タグ付けの結合を考慮した解析を行っている。しかし、この方法では、句構造解析の途中で各単語に与えられた品詞を参照する事が出来ない。

本稿では、この問題を解決するため、句構造解析の各規則で、単語に付与された品詞が参照可能となる解析手法を提案する。しかし、各規則で現在までに選択された品詞を保持する事は、文法サイズの増加を招く。CKY 法による句構造解析は、文法 G の規則数を $|G|$、対象とする文の単語数を $|\ell|$ としたとき、$O(|\ell|^3|G|)$ の計算量が必要な事から、文法サイズの増大により、効率的な解析が難しくなるという問題がある。そこで、本稿では、Coarse-to-Fine マルチパス探索法 [1, 7] によって、文法の階層構造を仮定し、その階層構造に基づいた探索空間の削減を行う事で、品詞タグ付けと句構造解析の同時学習及び予測を可能にする。文法の階層化としては、図 1 のように、句構造ラベルを階層的に簡略化する方法が最もよく知られている [1]。文献 [1] では生成モデルに基づく確率文脈自由文法 (PCFG) によって、句構造解析をモデル化しており、各レベルの文法を最尤推定によって学習している。そして、入力文が与えられると、低次モデルから順に文法規則の周辺確率を求め、その値がある閾値を下回る規則をフィルタリング (周辺化フィルタリング) して探索空間を削減する。しかし、この手法には次の課題が残されている。

- 周辺化フィルタリングは探索空間の削減効率が悪く、また、現在のモデルにおけるビタビ解析をフィルタリングしてしまう危険性がある [4][1]。
- 識別モデルに基づく PCFG [2] に比べ、解析精度が低い。

本稿で新たに提案する Coarse-to-Fine マルチパス探索法は、上記従来法の課題を解決するために、識別的

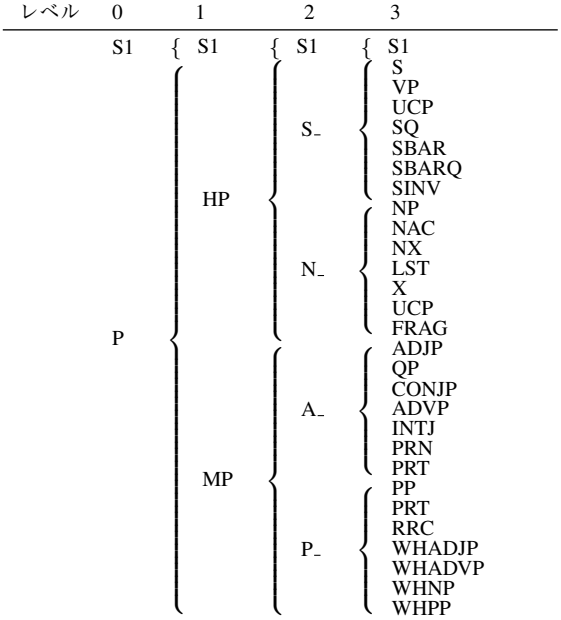

図 1: 英語 Penn Treebank における句構造ラベルの階層的簡略化

モデリングと新たな閾値基準に基づいて構成される。まず、提案法では、構造化パーセプトロン [5] によって、各規則に重みを与えた文脈自由文法に基づき、句構造解析をモデル化する。次に、フィルタリングでは、平均-最大閾値 [11] と呼ばれる基準を用い、低次モデルでは、その基準を満たすように識別学習することで、フィルタリング専用のモデル (**Max-marginal** フィルタモデル) を構築する。そして、削減された探索空間上で、今回新たに提案する、品詞タグ付けと句構造解析の同時学習を実施する。

実験の結果、文献 [2] 等で用いられた Penn Treebank の WSJ22,23 における文長 ≤ 15 の文[2] を対象とした実験設定において、提案手法は一般的な素性を用いて訓練された構文解析器に対して、10 分の 1 の探索空間で同等の精度が達成可能であることを確認した。さらに、品詞タグ付けとの同時解析による句構造解析の精度向上についても確認した。

[1] この課題を解決するため、文献 [4] では固定されたビタビ閾値に基づくフィルタリングを提案している。しかし、本稿の実験ではフィルタリングと学習を同時に行う必要があり、固定されたビタビ閾値では適切に文長の変化に対応する事が出来なかった。

[2] 今回、フィルタリングを行わない場合の評価も行う必要があり、その際の構文解析及び学習時間は図 5 のように膨大なものとなってしまうため、本稿においては 15 単語以下の文を用いて学習・評価を行った。

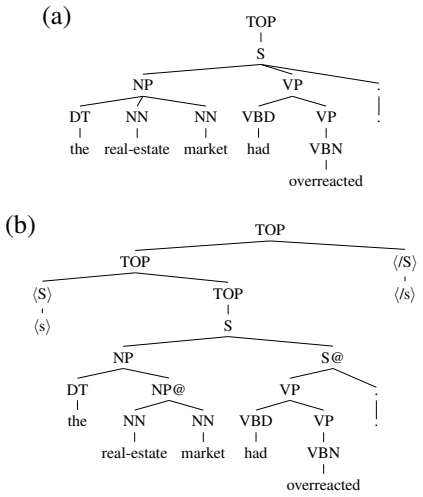

図 2: (a) の句構造木に前処理と右分岐 2 分化を施した結果 (b) (@は 2 分化後にできる特殊な句構造ラベルを表す。)

2 提案手法

2.1 文脈自由文法

文脈自由文法 G は組 $(N, PT, T, \text{TOP}, R)$ から成る。ここで N は非終端記号 (句構造ラベル) の集合、PT は前終端記号 (品詞タグ) の集合、T は終端記号 (単語) の集合、TOP は開始記号、R は生成規則の集合とする。本稿では $r \in R$ を次のような形に限定する。

- $\text{t} \rightarrow \text{x}$ ($\text{t} \in PT \wedge \text{x} \in T$),
- $\text{X} \rightarrow \text{Y}$ ($\text{X} \in N \wedge (\text{Y} \in (N \cup PT))$),
- $\text{X} \rightarrow \text{Y Z}$ ($\text{X} \in N \wedge (\text{Y, Z} \in (N \cup PT))$),
- $\text{TOP} \rightarrow \text{X}$ ($\text{X} \in N$).

また、特殊な開始終端記号 ⟨s⟩ と終了終端記号 ⟨/s⟩ に対して、次のような特殊な生成規則を R に追加する。

- ⟨S⟩→ ⟨s⟩,
- ⟨/S⟩→ ⟨/s⟩,
- TOP → ⟨S⟩ TOP,
- TOP → TOP ⟨/S⟩.

このような文法 G は図 2 のように、句構造木に対して 2 分化等の前処理を施した Penn Treebank コーパスから取得できる [3]。

2.2 品詞タグ付けと句構造解析の結合予測

入力文 $x = \text{x}_0 \ldots \text{x}_{\ell+1}$ ($\text{x}_0 = \langle\text{s}\rangle$, $\text{x}_{\ell+1} = \langle/\text{s}\rangle$, $\text{x}_1, \ldots, \text{x}_\ell \in T$) が与えられたとき、構文解析では各単語 x_i ($0 \leq i \leq \ell+1$) に規則 $\text{t} \rightarrow \text{x}_i \in R$ を割り当てる。演繹推論システム [10] では、この結果を $\overset{\text{t}}{\underset{i}{\triangle}}\,\underset{i}{}$ のような形式で表す。ここではこの形式で書かれたものを解析状態と呼ぶ。さらに、$\text{X} \rightarrow \text{Y Z} \in R$ のような規則を使って、2 つの解析状態 $\overset{\text{Y}}{\underset{i}{\triangle}}\,\underset{j}{}$ と $\overset{\text{Z}}{\underset{j+1}{\triangle}}\,\underset{k}{}$ から新たな解析状態 $\overset{\text{X}}{\underset{i}{\triangle}}\,\underset{k}{}$ を組み上げる。これは推論規則

$$\frac{\overset{\text{Y}}{\underset{i}{\triangle}}\,\underset{j}{} \qquad \overset{\text{Z}}{\underset{j+1}{\triangle}}\,\underset{k}{}}{\overset{\text{X}}{\underset{i}{\triangle}}\,\underset{k}{}} \; \text{X} \rightarrow \text{Y Z} \in R$$

で表される (他の形式の生成規則も同様に推論規則で記述可能)。推論規則に従って、終了となる解析状態 $\overset{\text{TOP}}{\underset{0}{\triangle}}\,\underset{\ell+1}{}$ に至ったとき、x に対して構文木が構築できることを意味する。CKY 法は最悪計算量 $O(|\ell|^3 \max(|PT|, |N|)^3)$ で構文解析を行う。

CKY 法で x を解析したとき、その導出過程で使われた生成規則の集合を R_x として書く。規則 $r \in R_x$ は R 中の規則とは異なり、$\text{X}_{i,k} \rightarrow \text{Y}_{i,j}\ \text{Z}_{j+1,k}$ のように各記号が対応した解析状態の情報 (ここではインデックス) が継承されて適用される範囲が限定されている。ただし、規則や規則集合に対する定義や関数は特に断わり無く、R_x にも適用できるものとする。R_x から構築できる構文木の集合を $\mathcal{Y}(R_x)$ として書く。このとき、$\mathcal{Y}(R_x)$ からより適切な構文木を取得するため、線形モデルを使って、各構文木を重み付けする。

$$\begin{aligned} \hat{y} &= \arg\max_{y \in \mathcal{Y}(R_x)} \mathbf{w} \cdot \mathbf{f}(x, y) \\ &= \arg\max_{y \in \mathcal{Y}(R_x)} \sum_{r \in y} \{\mathbf{w} \cdot \mathbf{f}(x, r)\}. \end{aligned} \tag{1}$$

ここで $\mathbf{f}(x, r)$ は、文 x と規則 r から抽出できる特徴量のベクトルを返す関数、$\mathbf{w}$ は特徴量を重み付けする重みベクトルを表す。構文木 $y \subseteq R_x$ は、その構築に使われた規則から成る集合である。品詞タグ付けでは、単語に割り当てられた品詞の連接を特徴量として考慮することができる。句構造解析で品詞の連接を特徴量として考慮するには、$\overset{\text{X}}{\underset{i}{\triangle}}\,\underset{j}{}$ のようにして、各解析状態にその左端及び右端の単語に割り当てられた品詞タグを記憶することが必要となる。図 3 にこの手法の演繹推論システムをまとめる。この構文解析は CKY 法を改良することで行うことができるが、最悪計算量は $O(|\ell|^3 \max(|PT|, |N|)^3 |PT|^4)$ となる。

2.3 Max-marginal フィルタ

図 1 で示したレベル i ($i \in \{0, 1, 2, 3\}$) のラベル定義に従って句構造木を簡略化し、文脈自由文法 $G_i = (N_i, PT, T, \text{TOP}, R_i)$ を構築することができる。ここでは、これらの文法に対して階層的な文法を用いた段階的なフィルタリングを行い、探索空間を削減する手法について説明する。

文法 G_i と入力文 x が与えられたとき、$R_{i,x}$ に対する規則フィルタ F を次のように定義する。

$$F(R_{i,x}) = \{r \in R_{i,x} \mid f(r, x, R_{i,x}) = 0\}. \tag{2}$$

ここで $f(r, x, R_{i,x})$ は、

$$f(r, x, R_{i,x}) = \begin{cases} 1 & \max_{y \in \mathcal{Y}(R_{i,x}) \wedge r \in y} \{\mathbf{w} \cdot \mathbf{f}(x, y)\} < t_x(\alpha) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

[3] 2 回以上連続する Unary 規則は、句構造ラベルを結合して 1 つの Unary 規則として扱った。また，Unary 規則が無い全ての箇所に左辺と右辺が同じ記号となる仮想的な規則を挿入した。

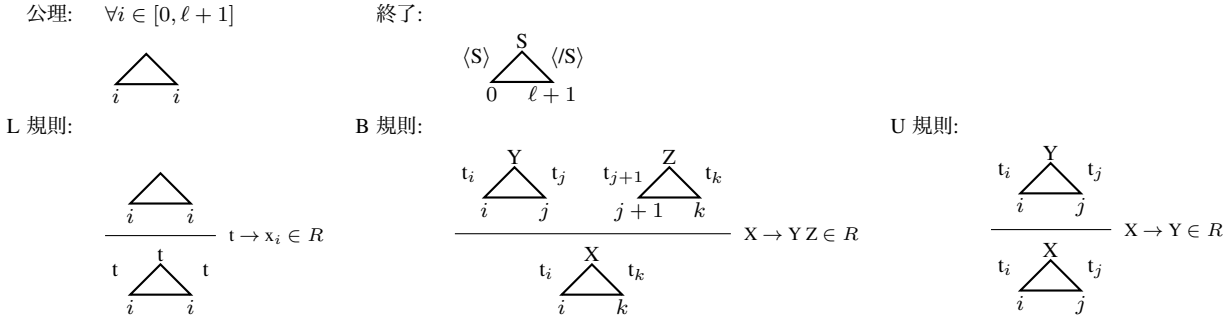

図 3: 入力文 $x = x_0 \ldots x_{\ell+1}$ に対する 1 次オーダ品詞タグ付けと句構造解析の同時解析を表す演繹推論システム

とする 2 値関数である。また、$t_x(\alpha)$ は、

$$t_x(\alpha) = \underbrace{\alpha \max_{y \in \mathcal{Y}(R_{i,x})} \{\mathbf{w} \cdot \mathbf{f}(x, y)\}}_{\text{ビタビスコア}} + \underbrace{\frac{(1-\alpha)}{|R_{i,x}|} \sum_{r \in R_{i,x}} \max_{y \in \mathcal{Y}(R_{i,x}) \wedge r \in y} \{\mathbf{w} \cdot \mathbf{f}(x, y)\}}_{\text{最大周辺化スコアの平均}} \quad (4)$$

として定義され、ビタビスコアと $R_{i,x}$ 中の全規則に対する最大周辺化スコアの平均をハイパーパラメータ $\alpha \in [0, 1)$ で結合した値である。$t_x(\alpha)$ は Inside-Outside 法を使って、効率的に求めることができる。この閾値を平均-最大閾値と呼び、これによる生成規則のフィルタを Max-marginal フィルタと呼ぶ。$\alpha = 1$ のとき、$F(R_{i,x}) = \hat{y}$ となる。$0 \le i < j \le 3$ とするとき、関数 $c_{i \to j} : N_i \cup PT \cup T \cup \{\text{TOP}\} \to 2^{N_j} \cup PT \cup T \cup \{\text{TOP}\}$ は N_i の要素である句構造ラベルに対し、図 1 の定義で対応する N_j の部分集合を返し、他の要素を引数にとるとそれをそのまま返す関数とする。これを用いて、規則 $r \in R_i$ を引数にとる関数

$$C_{i \to j}(r) = \{r' \mid r' = \begin{cases} \mathrm{X}' \ \to \ \mathrm{Y}'\,\mathrm{Z}' & \text{if } r = \mathrm{X} \ \to \ \mathrm{Y}\,\mathrm{Z} \\ \quad \text{where } \forall\, \mathrm{X}' \in c_{i \to j}(\mathrm{X}), \forall\, \mathrm{Y}' \in c_{i \to j}(\mathrm{Y}), & \\ \qquad \forall\, \mathrm{Z}' \in c_{i \to j}(\mathrm{Z}) & \\ \mathrm{X}' \ \to \ \mathrm{Y}' & \text{if } r = \mathrm{X} \ \to \ \mathrm{Y} \\ \quad \text{where } \forall\, \mathrm{X}' \in c_{i \to j}(\mathrm{X}), \forall\, \mathrm{Y}' \in c_{i \to j}(\mathrm{Y}) & \end{cases} \} \quad (5)$$

を定義する。さらに、規則集合 R_i を引数にとれるよう

$$C_{i \to j}(R_i) = \bigcup_{r \in R_i} C_{i \to j}(r) \quad (6)$$

として拡張する。提案手法は、構文木の集合

$$\mathcal{Y}(C_{2 \to 3}(F(C_{1 \to 2}(F(C_{0 \to 1}(F(R_{0,x})))))))) \quad (7)$$

から式 (1) の基準で最良の構文木を探索する。実験では G_0 によるフィルタよりも前に、線形モデルに基づいた点予測品詞タグ付けフィルタを設計して、単語辺りの品詞タグ数を削減している。また，各フィルタでは品詞の連接を考慮せず、通常の CKY 法及び Inside-Outside 法を使用する。

2.4 Max-marginal フィルタの学習

文法 G、及び、文とその正解構文木ペア (x, y) が与えられたとき、フィルタ損失を

$$L_f(x, y, R_x) = \begin{cases} 0 & \exists r \in y, f(r, x, R_x) = 0 \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

として定義する。また、フィルタ損失の凸上界として

$$\xi(x, y, R_x) = \max\{0, 1 - \mathbf{w} \cdot \mathbf{f}(x, y) + t_x(\alpha)\} \quad (9)$$

を定義する。

訓練データ $P = \{(x_1, y_1), \ldots, (x_{|P|}, y_{|P|})\}$ に対し、フィルタ損失を最小化するためのパラメータ $\mathbf{w}$ の学習は目的関数

$$\min_{\mathbf{w}} \frac{\lambda}{2} ||\mathbf{w}||^2 + \frac{1}{|P|} \sum_{p=1}^{|P|} \xi(x_p, y_p, R_{x_p}) \quad (10)$$

を最小化する問題として定式化することができる。ここで λ は正則化項を調節するハイパーパラメータである。これは確率的勾配降下法によって、

$$\mathbf{w}' \leftarrow (1 - \lambda)\mathbf{w} + \eta \mathbf{f}(x_p, y_p) - \eta\alpha \mathbf{f}(x_p, \hat{y}_p) - \frac{\eta(1-\alpha)}{|R_{x_p}|} \sum_{r \in R_{x_p}} \mathbf{f}(x_p, \operatorname*{arg\,max}_{y \in \mathcal{Y}(R_{x_p}) \wedge r \in y} \mathbf{w} \cdot \mathbf{f}(x, y)) \quad (11)$$

の更新式を用いてパラメータ $\mathbf{w}$ を最適化できる。ここで η は学習率である。

3 実験

マルチパス構文解析に基づき、品詞タグ付けと句構造解析の同時学習を行った際のフィルタリング精度と解析精度を評価するために実験を実施した。

精度比較のために、X-bar 文法を出力する基本となる構文解析器 (Basic) として、式 (10) と同様に、L2 正則化項を加えた構造化パーセプトロンに基づく構文解析器を実装した。学習データは文献 [2] の実験設定を参考に Penn Treebank の文長 ≤ 15 の文のみで構成される wsj15 を使用した。

スペースの都合により詳細は述べないが、Max-marginal フィルタモデルは品詞タグ付けにも応用できる。今回は句構造解析における L 規則を大幅に減らすことを目的とし、線形モデルに基づく点予測品詞タグ付けモデルをフィルタとして学習した。素性として、文献 [8] で述べられている素性のうち品詞の連接素性以外の全てを使用した。WSJ02-21 の文長 ≤ 15 のデータを使って学習を行い、WSJ22 の文長 ≤ 15 のデータを使ってテストした。各パラメータは $\lambda = 0.0$, $\eta = 1.0$、$\alpha = 0.95$ に設定した。文献 [11] に従い、フィルタ損失と効率損失の関係を求めた。表 1 に α を変えたときのフィルタ損失と効率損失を示す。以上の結果を元に、構文解析の実験では $\alpha = 0.9$ として推定した品詞タグ候補を入力として使用した。

構文解析器の素性は、文献 [3] の語彙素性を品詞タガーの 1-best で置換したものと文献 [2] で使用されて

$\alpha =$	0.7	0.8	0.9	1.0
フィルタ損失 (%)	0.28	1.08	2.25	5.15
効率損失 (%)	4.27	2.96	2.46	2.22

表 1: WSJ22 (長さ $\leq$ 15) に対する品詞タグ付けのフィルタ損失と効率損失

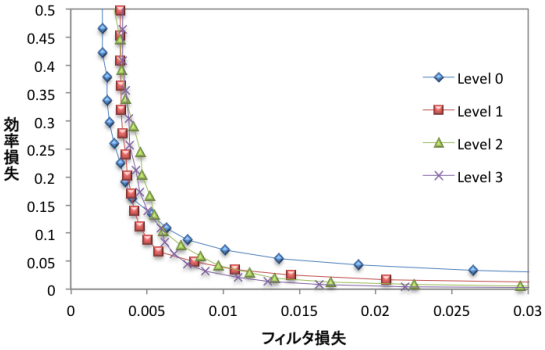

図 4: レベル 0–3 文法のフィルタリングの効率と損失

いる bigram 素性を用いた。品詞連接を考慮した解析を行う際には、語彙素性を品詞に置換したものを追加した。事前調査に基づき、学習時のハイパーパラメータは $\alpha = 1.0$、$\lambda = 0.2$、$\eta = 1.0$ に設定した。

品詞連接を考慮した構文解析は実行に膨大な計算量を要するため、学習の段階からフィルタリングを行う必要がある。文献 [1] において提案されているレベル 0–3 の文法を適切に用いてフィルタリングを行うために、上記で説明した構文解析器を使用し、フィルタ損失と効率損失の関係を求めた。結果を図 4 に示す。その結果、レベル 1, 2 の文法は、それぞれレベル 0, 3 の文法に似た性質を持っている事が判明したため、本実験においてはレベル 0, 3 の文法のみをフィルタリングに用いた。

構文解析精度の評価に際し、フィルタリングによる F 値の減少を明らかにするため、Basic をレベル 0 文法でフィルタリングし、探索空間を約 10 分の 1 にした場合 (Filtered) の F 値についても求めた。探索空間の計算には構文解析時に生成されたクリークの数を指標とし、WSJ22 の文長 $\leq$ 15 を使用した。品詞連接を考慮した解析 (Filtered + Pos) についてもレベル 0, 3 文法によるフィルタリングを行い、探索空間をそのまま解析する場合の約 100 分の 1 とした。さらに、基本となる構文解析器が正しく動いている事を確認するために、Berkeley Parser[6] と Transition based parser についても評価した。各設定における構文解析器の精度を表 2 に示す。

	WSJ22		WSJ23	
	BF (%)	PA (%)	BF (%)	PA (%)
Shift-Reduce	88.64	92.67	88.29	93.81
Berkeley	88.79	93.57	89.86	93.95
Basic	88.08	94.50	88.35	95.17
Filtered	87.89	94.45	88.01	95.23
Filtered + Pos	88.56	94.46	88.46	95.77

表 2: WSJ22 と 23 (長さ $\leq$ 15) に対するブラケット F 値 (BF) と品詞タグの正解率 (PA)

実験結果から、品詞と句構造の同時解析は、フィルタリングによって探索空間が大幅に減少した場合でも、F 値が増加している事が分かる。そして、品詞タグ付けの正解率も Basic に比べて上昇している。この数値に関しては、元々の品詞タガーの正解率が 95.75 であることから、品詞タガーに対する正解率の向上は確認

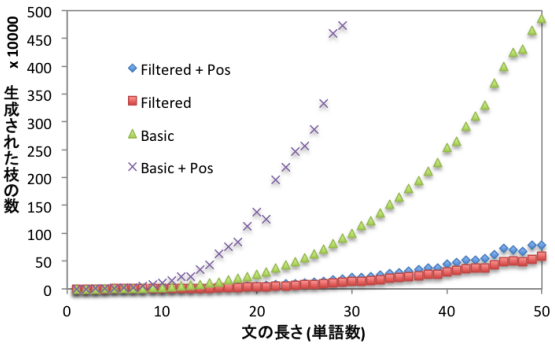

図 5: フィルタリングと探索空間

出来なかった。

最後に WSJ23 の全ての文を解析した際の文の長さと、生成される枝の数の平均の関係を図 5 に示す。Basic + Pos はフィルタリングを行わない場合の品詞と句構造の同時解析を表している。この結果から、フィルタリングを行わない場合には探索空間が爆発してしまうが、フィルタリングを行う事で大幅に探索空間を削減出来る事が分かる。またフィルタリングを段階的に行う事で、本来大きな計算量を持つ品詞連接を考慮した構文解析を行う際にも、探索空間の増加を抑える事が出来ることが分かった。

4 まとめ

本稿では、品詞と句構造解析の同時学習手法を提案し、また、それを可能とするための Max-Marginal フィルタリングの適用手法についても提案を行った。Penn Treebank を対象とした実験設定である wsj15 において、提案手法は精度を向上させながら探索空間を大幅に減らす事が可能である事を確認した。今後は今回実装した基本となる構文解析器の解析精度と速度を向上させ、より大きな規模のデータにおいて、他の構文解析器との比較を行う事を目的としたい。

参考文献

[1] E. Charniak, M. Johnson, M. Elsner, J. Austerweil, D. Ellis, I. Haxton, C. Hill, R. Shrivath, J. Moore, and M. Pozar. Multilevel coarse-to-fine pcfg parsing. In *Proc. of HLT-NAACL*, pp. 168–175, 2006.

[2] J. R. Finkel, A. Kleeman, and C. D. Manning. Efficient, feature-based, conditional random field parsing. In *Proc. of ACL*, pp. 959–967, 2008.

[3] D. Hall, G. Durrett, and D. Klein. Less grammar, more features. In *Proc. of ACL*, pp. 228–237, 2014.

[4] L. Huang. Forest reranking: Discriminative parsing with non-local features. In *Proc. of ACL*, pp. 586–594, 2008.

[5] M.Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of ACL*, pp. 1–8, 2002.

[6] S. Petrov, L. Barrett, R. Thibaux, and D. Klein. Learning accurate, compact, and interpretable tree annotation. In *Proc. of ACL*, pp. 433–440, 2006.

[7] S. Petrov and D. Klein. Improved inference for unlexicalized parsing. In *Proc. of HLT-NAACL*, pp. 404–411, 2007.

[8] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*, Vol. 1, pp. 133–142, 1996.

[9] A. M. Rush, D. Sontag, M. Collins, and T. Jaakkola. On dual decomposition and linear programming relaxations for natural language processing. In *Proc. of EMNLP*, pp. 1–11, 2010.

[10] S. Shieber, Y. Schabes, and F. Pereira. Principles and implementation of deductive parsing. *The Journal of logic programming*, Vol. 24, No. 1, pp. 3–36, 1995.

[11] D. Weiss and B. Taskar. Structured prediction cascades. In *Proc. of AISTATS*, 2010.