

オンライン最大マージン学習アルゴリズムに基づく 多言語依存構造-述語項構造解析

渡邊 陽太郎^{†1} 浅原 正幸^{†1} 松本 裕治^{†1}

本稿では、オンライン最大マージン学習アルゴリズムに基づく依存構造-述語項構造解析システムについて述べる。本システムは、二次素性を用いた交差無し依存構造解析器および大域的素性を用いた述語項構造解析器により構成される。双方の解析器の学習には、オンライン最大マージン学習アルゴリズムの一種である Passive-Aggressive アルゴリズム⁴⁾を用いた。また、述語項構造解析器には、単一のモデルにより局所的素性と大域的素性の双方を扱うことが可能な学習・解析手法を適用した。多言語の依存構造-述語項構造解析が課題となっていた CoNLL-2009 Shared Task⁸⁾において、本システムは、英語で2位、チェコ語で3位という結果が得られた。

Multilingual Syntactic-Semantic Dependency Parsing Using Online Large-Margin Learning Algorithms

YOTARO WATANABE,^{†1} MASAYUKI ASAHARA^{†1}
and YUJI MATSUMOTO^{†1}

This paper describes a system for syntactic-semantic dependency parsing for multiple languages. The system consists of two parts: a state-of-the-art higher-order projective dependency parser for syntactic dependency parsing, and a predicate-argument structure analyzer with global features. All components are trained with the Passive-Aggressive Algorithm⁴⁾, which is an online large-margin learning algorithm. Also, in predicate-argument structure analysis, we propose a novel margin-based learning algorithm that can deal with both local features and global features. In the closed challenge of the CoNLL-2009 Shared Task⁸⁾, our system achieved the 2nd best performance for English, and the 3rd best performance for Czech.

^{†1} 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

1. はじめに

単語または文節間の修飾関係の同定により統語構造を得る依存構造解析と、文中の述語および述語の項、またそれぞれの項が持つ意味役割を同定する述語項構造解析は自然言語処理の上位アプリケーションのための中間的な技術として必要不可欠である。既に、統計的機械翻訳、質問応答などの応用分野において、双方の情報を利用することの有効性について報告されている^{28),33)}。

依存構造および述語項構造解析のための言語資源は、多くの言語において整備が進んでいる。まず英語に関して、依存構造解析では句構造木から依存構造木への変換のための主辞規則に関する研究¹²⁾の進展があり、動詞述語の項構造は Proposition Bank Project²⁶⁾、名詞述語の項構造は NomBank Project²²⁾によりコーパスの整備が進められた。日本語に関しては、京都テキストコーパスや NAIST テキストコーパス³⁵⁾があり、また他の言語についても、チェコ語、ドイツ語、中国語など、多くの言語において同様の情報が付与されたコーパスの整備が進んでいる。こうした背景の中で、特定の言語に依存しない、汎用的で頑健な依存構造および述語項構造解析の枠組みが求められており、高い精度での双方のタスクの解析手法に研究者の関心が集まっている。そこで本研究では、言語に非依存で頑健な依存構造および述語項構造解析システムの開発を目的とする。

依存構造および述語項構造解析の手法としては近年、双方の問題が依存した形で学習をおこなう結合学習 (Joint Learning) の枠組みに研究者の関心が集まっている^{8),29)}。しかし、これらのタスクにおいては結合学習の優位性は認められていない。そこで本研究では個々の解析の性能の向上に注力することにする。

依存構造解析の分野においては、係り受け関係のスコアが独立に決定される一次の依存構造解析アルゴリズムが主流であったが、近年になり、ある特定の係り受け関係が他の係り受け関係のスコアに影響を与えるような高次の依存構造解析アルゴリズムの研究が進展し、一次の解析アルゴリズムに対する優位性が示されている^{2),19),21)}。その優位性を享受するために、本システムでは高次依存構造解析アルゴリズムの一つである二次交差なし依存構造解析アルゴリズム²⁾を本システムの依存構造解析器に用いる。

述語項構造解析の分野では、N-best 解の中から最適な解を大域的な情報を捉えた上で選択するリランキングを用いた手法の有効性が示されている^{13),31)}。一方、文献 15) では、局所的素性と大域的素性を単一のモデルで扱う手法が、リランキングと比較して良い結果が得られたと報告している。我々はこの結果に注目し、述語項構造解析を構造予測問題として定式

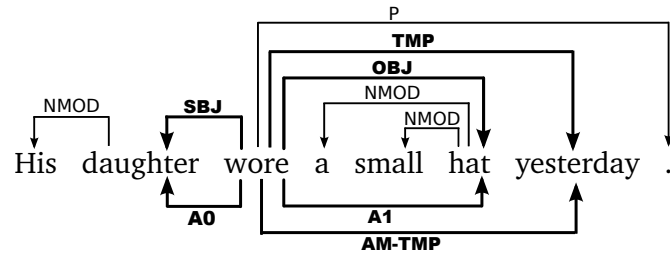


図1 依存構造木に基づく述語項構造解析の例(英語)

化し、単一のモデルで双方の素性を扱う手法を適用する。文献15)では、大域的素性を扱うモデル学習のためのアルゴリズムとしてパーセプトロンのみ示されているが、パーセプトロンと比較して性能の良い学習アルゴリズムがいくつか存在する。本研究では、その中の一つである Passive-Aggressive アルゴリズム⁴⁾を用いて、局所的素性と大域的素性の双方を用いたモデルの学習を試みる。

2. 依存構造に基づく述語項構造解析

依存構造木に基づく英語の述語項構造解析の例を図1に示す。文の上側に示されている矢印は統語的な依存関係、下側に示されている矢印は述語“wore”の述語項構造を表している。この例において、述語“wore”に対して意味役割を持つ項は、“His daughter”(動作主)、“a small hat”(対象)、“yesterday”(時間)である。英語の PropBank²⁶⁾ や NomBank²²⁾ の場合、意味役割のラベルは数値化されており、必須の意味役割については、動作主が A0、対象や被動作主は A1、その他述語ごとに異なる意味役割が A2 から A5 までのラベルに対応付けられており、解析時にはこれらのラベルを付与する。また、述語が要求する必須の項以外に、時間 (AM-TMP)、場所 (AM-LOC) といった付加詞の役割を持つ項についても、文中に出現した場合は同定する。

句構造木に基づく意味役割付与タスクでは、“His daughter”のスパンに対応する句構造木上のノードに対して A0 を、“a small hat”のスパンに対応する句構造木上のノードに対して A1 を付与する。一方、依存構造木に基づく意味役割付与タスクでは、ある特定の意味役割を持つスパン内の語を全て支配している主辞に対して意味役割を付与する問題として扱われる。例えば、図1の例では、句“His daughter”の“His”は“daughter”を修飾しているため句の主辞は daughter である。よって、“daughter”に対して A0 を付与する。また、句“a small

hat”の“a”および“small”は“hat”を修飾しているため、句“a small hat”の主辞は“hat”である。よって“hat”に対して A1 を付与する。

3. オンライン最大マージン学習アルゴリズム

本システムの全ての解析器は、オンライン最大マージン学習アルゴリズムの一つである Passive-Aggressive アルゴリズム⁴⁾を用いて学習をおこなう。このアルゴリズムは、事例 (x, y) ごとに以下の制約付き最適化問題を解くことによりパラメータの更新をおこなう。この最適化問題では、制約 $l(w; (x, y)) \leq \xi$ を満たし、なおかつ更新前のパラメータ w からの変化が最小となるような更新後のパラメータ w_{new} を得る。

$$w_{new} = \arg \min_{w' \in \mathbb{R}^n} \frac{1}{2} \|w' - w\|^2 + C\xi \quad \text{s.t. } l(w; (x, y)) \leq \xi \text{ and } \xi \geq 0 \quad (1)$$

w はパラメータベクトル、 w_{new} は更新後のパラメータベクトル、 ξ はソフトマージンのためのスラック変数、 C は ξ の影響をコントロールする正の定数である。 $l(w; (x, y))$ は損失関数であり、構造予測問題では以下のような損失関数が用いられる。

$$l(w; (x, y)) = w \cdot \Phi(x, \hat{y}) - w \cdot \Phi(x, y) + \rho(y, \hat{y}) \quad (2)$$

$\rho(y, \hat{y})$ は、正解の構造 y と現在の学習器の予測 \hat{y} の構造の違いにより与える損失であり、この値が学習器にマージンを確保させる役割を果たす。

この最適化は閉形式解として得られ、以下のような更新則が得られる。

$$\tau = \min \left(C, \frac{w \cdot \Phi(x, \hat{y}) - w \cdot \Phi(x, y) + \rho(y, \hat{y})}{\|\Phi(x, y) - \Phi(x, \hat{y})\|^2} \right) \quad (3)$$

$$w_{new} = w + \tau (\Phi(x, y) - \Phi(x, \hat{y})) \quad (4)$$

文献4)では、上記の更新式を PA-I とし、(1)式のスラック変数 ξ を除いた PA、 ξ が ξ^2 に置き換えられた PA-II を加えた3種類の学習アルゴリズムが提案されている。本研究では全ての解析器の学習に PA-I を用いる。

パーセプトロンをはじめとするオンライン学習アルゴリズムでは、学習時に与えるデータの順序により得られるパラメータは変化する。与えるデータの順序に起因するパラメータの分散を抑えるために、パラメータの平均化がしばしばおこなわれる。ここで、パーセプトロンや Passive-Aggressive アルゴリズムといった、ベクトルにある係数を掛けて足し込むタイプ (additive) のアルゴリズムでは、効率的なパラメータの平均化が可能である。文献5)の方法による平均化 Passive-Aggressive アルゴリズムを図2に示す。

```

input Training set  $\mathcal{T} = \{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^T$ , Number of iterations N and Parameter C
 $\mathbf{w} \leftarrow \mathbf{0}$ ,  $\mathbf{w}_a \leftarrow \mathbf{0}$ ,  $c \leftarrow 1$ 
for  $i \leftarrow 0$  to N do
  for  $(\mathbf{x}_t, \mathbf{y}_t) \in \mathcal{T}$  do
     $\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} \mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathbf{y}) + \rho(\mathbf{y}_t, \hat{\mathbf{y}})$ 
     $\tau_t = \min \left( C, \frac{\mathbf{w} \cdot \Phi(\mathbf{x}_t, \hat{\mathbf{y}}) - \mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathbf{y}_t) + \rho(\mathbf{y}_t, \hat{\mathbf{y}})}{\|\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}})\|^2} \right)$ 
     $\mathbf{w} \leftarrow \mathbf{w} + \tau_t (\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}))$ 
     $\mathbf{w}_a \leftarrow \mathbf{w}_a + c \tau_t (\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}))$ 
     $c \leftarrow c + 1$ 
  end for
end for
return  $\mathbf{w} - \mathbf{w}_a / c$ 

```

図 2 平均化 Passive-Aggressive アルゴリズム

4. 二次依存構造解析アルゴリズム

これまでに提案されている依存構造解析アルゴリズムは、グラフベースの手法では交差なしの依存構造木を得る Eisner アルゴリズム⁶⁾、交差の有る依存構造木を得る最大全域木アルゴリズム²⁰⁾、遷移ベースの手法では Shift-Reduce 法に基づくアルゴリズム²³⁾がある。グラフベースの手法は単純なモデル化の場合、依存構造木のスコアは $s(\mathbf{y}) = \sum_{(h,m) \in \mathbf{y}} \mathbf{w} \cdot \Phi(h, m, \mathbf{x})$ といった形で各主辞と修飾語のペアに因子化 (factorize) し、このスコアが最大となるような依存構造木を求める問題として扱われる。これは一次の依存構造解析と呼ばれる。

一方、近年注目されているのは、ある修飾関係が、他の修飾関係に依存してスコアが決定されるような二次依存構造解析アルゴリズムであり、一次の依存構造解析アルゴリズムと比較して高い精度が得られた事が報告されている^{2),19),21)}。交差なしの場合は Eisner アルゴリズムを拡張したアルゴリズム^{2),19)}、交差ありの場合は二次最大全域木アルゴリズム²¹⁾などがある。

本システムの依存構造解析器には、文献 2) の二次交差なし依存構造解析アルゴリズムを用いる。まず、以下のようなスコア関数を定義する。

$$s(\mathbf{y}) = \sum_{(h,m) \in \mathbf{y}} F(h, m, \mathbf{x}) \quad (5)$$

ここで、 h は主辞、 m は修飾語のペアで、依存構造木 \mathbf{y} のある特定の修飾関係を表している。また、 $F(h, m, \mathbf{x})$ は、一つの依存構造の辺 (h, m) に関してスコアを定義する因子 (factor) で

表 1 依存構造解析器に用いた素性

一次	Unigram	hl, hp, hl+hp, dl, dp, dl+dp
	Bigram	hp+dp, hl+dl, hl+dl, hl+hp+dl, hl+hp+dp, hl+dl+dp, hp+dl+dp, hl+hp+dl+dp
	文脈	hp+hp+1+dp-1+dp, hp-1+hp+dp-1+dp, hp+hp+1+dp+dp+1, hp-1+hp+dp+dp+1
	距離	主辞と修飾語の間にある単語数
二次	Trigram	hl+cl, hl+cl+cp, hp+cl, hp+cp, hp+dp+cp, dp+cp, dp+cl+cp, dl+cp, dl+cp+cl

あり、この因子の値の和が依存構造木全体のスコアとなる。

$F(h, m, \mathbf{x})$ は、文献 2) と同様、二次の因子化によるスコアリングをおこなう。具体的には、関数 $F(h, m, \mathbf{x})$ を以下のように定義する。

$$F(h, m, \mathbf{x}) = \mathbf{w} \cdot \{ \Phi(h, m, \mathbf{x}) + \Phi(h, m, c_h, \mathbf{x}) + \Phi(h, m, c_{mi}, \mathbf{x}) + \Phi(h, m, c_{mo}, \mathbf{x}) \} \quad (6)$$

\mathbf{w} はパラメータベクトル、 Φ は素性ベクトル、 c_h は、スパン $[h...m]$ 内の h の子ノードで m に最も近いノード、 c_{mi} はスパン $[h...m]$ 内の m の子ノードで m から最も遠いノード、 c_{mo} はスパン $[h...m]$ の外にある m の子ノードのうち m から最も遠いノードを表している。二次交差なし依存構造解析アルゴリズムの詳細に関しては、文献 2) を参照されたい。

解析器の学習には、図 2 の平均化 Passive-Aggressive アルゴリズムを用いた。 $\rho(\mathbf{y}_t, \hat{\mathbf{y}})$ の値は、 $\rho(\mathbf{y}_t, \hat{\mathbf{y}}) = \sum_{(y_t, \hat{y}) \in (\mathbf{y}_t, \hat{\mathbf{y}})} \rho(y_t, \hat{y})$ という形で依存構造木の辺ごとに分解して設定した。 $\rho(y_t, \hat{y})$ の値は、 $y_t \neq \hat{y}$ であれば 1.0、正しい場合は 0.0 とした。

依存構造解析器に用いる素性は、基本的には文献 11) で使用されているものに加えて、距離素性を追加した素性集合を使用した。ここで、素性の表現を簡略化するために、以下の略記を用いることにする。'h' を主辞、'd' を修飾語、'c' を主辞または修飾語の子、'l' を見出し語、'p' を述語、'-1' をある単語の一つ左の位置、'+1' をある単語の一つ右の位置とする。表 1 に依存構造解析器に用いた素性の一覧を示す。

5. 大域的素性を用いた述語項構造解析

本稿では、述語項構造解析を、述語の語義の同定と、述語に対する項の意味役割の同定 (意味役割付与) の二つを統合したタスクとして定義する。本システムの述語項構造解析システムは、述語語義同定器と意味役割同定器の二つの解析器から構成され、これらはパイプラインに処理がおこなわれる。

5.1 述語の語義曖昧性解消

本システムにおける述語項構造解析の最初のプロセスは、述語の語義曖昧性解消である。述語の語義曖昧性解消は、各述語がどの語義を持つかを判定する多クラス分類問題として扱うことができる。そこで、各見出し語ごとに線形モデルを用意し、平均化 Passive-Aggressive

表 2 述語の語義曖昧性解消に用いた素性

単語素性	述語および述語の主辞の見出し語, 述語の品詞, またそれらの組み合わせ.
依存構造ラベル	述語と述語の主辞の間の依存構造ラベル.
依存構造ラベル系列	述語に支配されている子ノードの依存構造ラベルを, 単語の出現順に結合した系列.

アルゴリズムを用いて語義の分類器を作成した. 分類に使用した素性を表 2 に示す.

述語の語義選択に有効な素性は言語ごとに異なると考えられる. そこで, 各言語において最適な素性集合を得るために前向き素性選択法を適用した.

5.2 意味役割割付与

意味役割割付与では, 局所的素性と大域的素性の双方を単一のモデルで扱う枠組みを採用し, 双方の素性を単一のモデルで学習が可能なオンライン最大マージン学習アルゴリズムを適用する.

5.2.1 大域的素性を用いた意味役割割付与モデル

$\mathcal{A}(p) = \{a_1(p), a_2(p), \dots, a_N(p)\}$ を, 述語 p が与えられた時の項候補に対する意味役割ラベルの割当とする. 例えば, 図 1 の例で述語を $p = \text{wore}$ とすると, $\mathcal{A}(\text{wore}) = \{\text{“His”}=\text{NONE}, \text{“daughter”}=\text{A0}, \text{“wore”}=\text{NONE}, \text{“a”}=\text{NONE}, \text{“small”}=\text{NONE}, \text{“hat”}=\text{A1}, \text{“yesterday”}=\text{AM-TMP}, \text{“.”}=\text{NONE}\}$ といった割当のことをいう.

意味役割与モデルにて用いる $\mathcal{A}(p)$ に対するスコア関数 $s(\mathcal{A}(p))$ を, 依存構造解析と同様に因子の和 $s(\mathcal{A}(p)) = \sum_k F_k(\mathbf{x}, \mathcal{A}(p))$ とする. ここで, 二種類のスコア関数: 局所的因子 $F_L(\mathbf{x}, \mathcal{A}(p)) = \sum_{a(p) \in \mathcal{A}(p)} \mathbf{w} \cdot \Phi_L(\mathbf{x}, a(p))$ と大域的因子 $F_G(\mathbf{x}, \mathcal{A}(p)) = \mathbf{w} \cdot \Phi_G(\mathbf{x}, \mathcal{A}(p))$ を導入する. Φ_L は局所的因子に対する素性関数であり, F_L は各項候補がどのような意味役割を持つかを個別に捉える. Φ_G は大域的因子に対する素性関数であり, F_G は \mathcal{A} がどのような大域的構造を持っているかを捉える. これらの関数の導入により, 結果として得られる $\mathcal{A}(p)$ のスコア関数は以下ようになる. このスコアが最大となる割当を, 最適なものとして選択する.

$$s(\mathcal{A}(p)) = \sum_{a(p) \in \mathcal{A}(p)} \mathbf{w} \cdot \Phi_L(\mathbf{x}, a(p)) + \mathbf{w} \cdot \Phi_G(\mathbf{x}, \mathcal{A}(p)) \quad (7)$$

5.3 最適な割当の計算

一般に, 大域的素性を用いた場合, 最適な割当 (argmax) の効率的な計算は困難である. 正確な解を求めたい場合, 全ての可能な割当を列挙した上で大域的素性のスコアを足し, その中で最もスコアの高い割当を選択するという処理が必要となるが, それは非効率的であり現実的な選択肢とは言えない. ここでは, 文献 36) と同様の方法により最適な割当を求めるこ

```

input Training set  $\mathcal{T} = \{\mathbf{x}_t, \mathcal{A}_t\}_{t=1}^T$ , Number of iterations  $N$  and Parameter  $C$ 
 $\mathbf{w} \leftarrow \mathbf{0}, \mathbf{v} \leftarrow \mathbf{0}, c \leftarrow 1$ 
for  $i \leftarrow 0$  to  $N$  do
  for  $(\mathbf{x}_t, \mathcal{A}_t) \in \mathcal{T}$  do
    let  $\Phi_a(\mathbf{x}_t, \mathcal{A}) = \sum_{a \in \mathcal{A}} \Phi_L(\mathbf{x}_t, a) + \Phi_G(\mathbf{x}_t, \mathcal{A})$ 
    generate  $N$ -best assignments  $\{\mathcal{A}^n\}$  using  $F_L$ 
     $\hat{\mathcal{A}} = \arg \max_{\mathcal{A} \in \{\mathcal{A}^n\}} \mathbf{w} \cdot \Phi_a(\mathbf{x}_t, \mathcal{A}) + \rho(\mathcal{A}_t, \mathcal{A})$ 
    if  $\hat{\mathcal{A}} \neq \mathcal{A}_t$  then
       $\tau_t = \min \left( C, \frac{\mathbf{w} \cdot \Phi_a(\mathbf{x}_t, \hat{\mathcal{A}}) - \mathbf{w} \cdot \Phi_a(\mathbf{x}_t, \mathcal{A}_t) + \rho(\mathcal{A}_t, \hat{\mathcal{A}})}{\|\Phi_a(\mathbf{x}_t, \mathcal{A}_t) - \Phi_a(\mathbf{x}_t, \hat{\mathcal{A}})\|^2} \right)$ 
       $\mathbf{w} \leftarrow \mathbf{w} + \tau_t (\Phi_a(\mathbf{x}_t, \hat{\mathcal{A}}) - \Phi_a(\mathbf{x}_t, \mathcal{A}_t))$ 
       $\mathbf{w}_a \leftarrow \mathbf{w}_a + c \tau_t (\Phi_a(\mathbf{x}_t, \hat{\mathcal{A}}) - \Phi_a(\mathbf{x}_t, \mathcal{A}_t))$ 
    else if  $\hat{\mathcal{A}} = \mathcal{A}_t$  and  $\mathcal{A}^1 \neq \mathcal{A}_t$  then
       $\gamma_t = \min \left( C, \frac{\mathbf{w} \cdot \Phi_L(\mathbf{x}_t, \mathcal{A}^1) - \mathbf{w} \cdot \Phi_L(\mathbf{x}_t, \mathcal{A}_t) + \rho(\mathcal{A}_t, \mathcal{A}^1)}{\|\Phi_L(\mathbf{x}_t, \mathcal{A}_t) - \Phi_L(\mathbf{x}_t, \mathcal{A}^1)\|^2} \right)$ 
       $\mathbf{w} \leftarrow \mathbf{w} + \gamma_t (\Phi_L(\mathbf{x}_t, \mathcal{A}^1) - \Phi_L(\mathbf{x}_t, \mathcal{A}_t))$ 
       $\mathbf{w}_a \leftarrow \mathbf{w}_a + c \gamma_t (\Phi_L(\mathbf{x}_t, \mathcal{A}^1) - \Phi_L(\mathbf{x}_t, \mathcal{A}_t))$ 
    end if
     $c \leftarrow c + 1$ 
  end for
end for
return  $\mathbf{w} - \mathbf{w}_a/c$ 

```

図 3 大域的素性を用いた意味役割割付与モデルの平均化 Passive-Aggressive アルゴリズムによる学習

とにする.

$$\{\mathcal{A}(p)^n\} = N\text{best}_{\mathcal{A}(p)} \sum_{a(p) \in \mathcal{A}(p)} \Phi_L(\mathbf{x}_t, a(p)) \quad (8)$$

$$\hat{\mathcal{A}}(p) = \arg \max_{\mathcal{A}(p)' \in \{\mathcal{A}(p)^n\}} \sum_{a(p) \in \mathcal{A}(p)} \mathbf{w} \cdot \Phi_L(\mathbf{x}, a(p)) + \mathbf{w} \cdot \Phi_G(\mathbf{x}, \mathcal{A}(p)) \quad (9)$$

まず最初に局所的素性のみを用いて意味役割ラベルの割当の N -best 解 $\{\mathcal{A}(p)^n\}$ を得る. 次に, 各割当に対して大域的素性のスコアを足し, その中で最もスコアの高い割当 $\hat{\mathcal{A}}(p)$ を選択する. 本システムでは, N -best 解の生成にビームサーチアルゴリズムを用いた.

5.3.1 モデルの学習

文献³⁶⁾の大域的素性を用いたパーセプトロン学習の特徴は以下の 2 点に集約される.

- (1) $\hat{y} \neq y_t$ の場合, Φ_a が正解を出力するように学習をおこない, 正解と二次解の間にマージンが確保されるようにする
- (2) $\hat{y} = y_t$ and $y^1 \neq y_t$ の場合, Φ_L が正解を出力するように学習をおこない, y_t と y^2

表 3 意味役割付与モデルに用いた素性

局所的素性	述語, 述語の親, 項候補, 項候補の親, 項候補の最左, 最右の子, 最左, 最右の兄弟の見出し語と品詞 述語, 項候補, 項候補の子の依存構造ラベル 述語と項の間の親子関係 (例: 子 or 兄弟 or etc.) 述語と項の前後関係 (例: arg-pred or pred-arg) 述語の子の依存構造ラベルを出現順序順に並べた系列 依存構造上の述語と項の間のパス (見出し語, 品詞, 依存構造ラベル) 述語と項の間にある依存構造の辺の数
大域的素性	述語の語義と意味役割を, 対応する単語の出現順序順に並べた系列 (例: A0-wear.01-A1) フレームに定義されている意味役割が含まれているか (例: CONTAINS:A1 or MISSING:A1)

の間にマージンが確保されるようにする (y^1, y^2 は, 局所的素性のみを用いて得られた一次解, 二次解とする)

(2) の更新が必要な理由は, (1) の更新のみでは, Φ_L のみを用いたスコアに, 正解とそれ以外の間のマージンが確保される保証が無いためである. このマージンが確保されないと, N-best 解の中に正解が含まれない可能性が高くなる. その可能性をできるだけ排除するため, Φ_L のみを用いても正解を出力するように (2) の更新をおこなう.

この学習を Passive-Aggressive アルゴリズムにておこなう場合, 条件 (1), (2) それぞれにおいて, 以下の最適化問題を解くことに対応する.

$$\mathbf{w}_{new} = \arg \min_{\mathbf{w}' \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{w}' - \mathbf{w}\|^2 \begin{cases} \text{s.t. } l_a(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = 0 & \hat{\mathbf{y}} \neq \mathbf{y} \\ \text{s.t. } l_l(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = 0 & \hat{\mathbf{y}} = \mathbf{y} \text{ and } y^1 \neq \mathbf{y} \end{cases} \quad (10)$$

ここで, 損失 l_a および l_l は以下のようになる.

$$l_a(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w} \cdot \Phi_a(\mathbf{x}, \hat{\mathbf{y}}) - \mathbf{w} \cdot \Phi_a(\mathbf{x}, \mathbf{y}) + \rho(\mathbf{y}, \hat{\mathbf{y}}) \quad (11)$$

$$l_l(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w} \cdot \Phi_l(\mathbf{x}, \hat{\mathbf{y}}) - \mathbf{w} \cdot \Phi_l(\mathbf{x}, \mathbf{y}) + \rho(\mathbf{y}, \hat{\mathbf{y}}) \quad (12)$$

結果として得られる学習アルゴリズムを図 3 に示す. 本システムの意味役割同定器の学習では, $\rho(\mathcal{A}_t, \hat{\mathcal{A}})$ の値は, $\rho(\mathcal{A}_t, \hat{\mathcal{A}}) = \sum_{(a_t, \hat{a}) \in (\mathcal{A}_t, \hat{\mathcal{A}})} \rho(a_t, \hat{a})$ のように依存構造解析のものと同様に因子化し, $\rho(a_t, \hat{a})$ は $a_t \neq \hat{a}$ であれば 1.0, 正解であれば 0.0 に設定した.

意味役割付与モデルの学習および解析に用いた素性を表 3 に示す. 局所的素性は文献 32) と同様のものを, 大域的素性は文献 13) の英語の述語項構造のリランキングに用いられた素性を基にした.

5.3.2 項候補の枝狩り

依存構造木上での意味役割付与においては, 正解の項は述語の周辺にあることが多く, その傾向を利用する事により項候補の枝狩りをおこなうことが可能である. 図 1 の英語の例

表 4 各言語における項候補の枝狩りの結果. 被覆率は, 意味役割を持つ正解の項が含まれている割合を, 項候補の削減率は, 枝狩り前と枝狩り後の項候補全体数の比率を示している.

	Catalan	Chinese	Czech	English	German	Japanese	Spanish
枝狩りアルゴリズム	1	1	2	1	1	品詞	1
被覆率 (%)	100	98.9	98.5	97.3	98.3	99.9	100
項候補の削減率 (%)	69.1	69.1	49.1	63.1	64.3	41.0	69.7

では, 述語 “wore” の項 (“daughter” および “hat”) は, 依存構造木上では “wore” の直接の子になっている. このように能動態の場合は, 項となる句の主辞が直接, 述語に支配されるような構造を持っているため, 述語の子のみを項候補とすればよいことになる. 受動態の場合や従属節内の述語の項など, 文の構造によっては必ずしも述語の直接の子になるわけではないが, 英語の場合, 基本的にはある述語から ROOT まで辿って行く間に通るノードの子を候補とするだけで殆どの項をカバーする事が可能である. この枝狩りの処理は, 探索空間の削減による解析時間の高速化と, 過剰な負例による再現率の低下といった学習への悪影響を抑制する効果を与える.

各言語について述語と項の位置関係について調査したところ, 言語ごとの傾向はやや異なっていた. そこで, 以下に述べる 2 種類の項候補の枝狩りアルゴリズムを適用した. これらのアルゴリズムによって得られた要素のみを, 意味役割を付与する対象となる項候補として扱う.

5.3.2.1 項候補の枝狩りアルゴリズム 1:

S を項候補集合とし, n を現在のノード, n の親ノードを得る関数を $h(n)$, 現在注目している述語のノードを p とする. 初期化: $S \leftarrow \phi$ とし, 述語のノードを開始点とする ($n \leftarrow p$).

(1) 現在のノードの子ノードを S に全て追加する. (2) 現在のノードの親ノードを現在のノードとする ($n \leftarrow h(n)$). (3) (1)-(2) を, $n = \text{ROOT}$ となるまで繰り返す.

5.3.2.2 項候補の枝狩りアルゴリズム 2:

枝狩りアルゴリズム 1 と基本的には同一であるが, S に追加するノードを, 現在のノードの子ノードだけでなく, 孫ノードも含める.

項候補の枝狩りの結果を表 4 に示す. 日本語に関しては, 上記のアルゴリズムでは高い被覆率が得られなかったため, 単純に品詞による項候補の枝狩りをおこなった.

6. 実験

CoNLL-2009 Shared Task データを用いて実験をおこなった. このデータは, カタロニア

語、中国語、チェコ語、英語、ドイツ語、日本語、スペイン語の7言語のデータから構成され、それぞれのデータには依存構造および述語項構造の情報が付与されている。英語のデータは、Penn Treebank に PropBank と NomBank の情報を付与し、文献 12) で述べられている主辞規則により句構造木から依存構造木へ変換したものである。日本語のデータは、京大コーパスのうち、格要素の情報が付与された文節単位のデータを、形態素単位の依存構造木へ変換したものである。他の言語のデータの詳細については、文献 8) を参照されたい。

評価指標は、以下のうち太字で記述された3種類を用いる。

$$\text{Labeled Syntactic Accuracy (LSA)} = \frac{\text{係り先とラベルの両方が正解のエッジ数}}{\text{依存構造の全辺数}}$$

$$\text{Semantic Labeled Precision (SLP)} = \frac{\text{述語語義同定正解数} + \text{意味役割付与正解数}}{\text{解析器が述語と判定した総数} + \text{解析器が項と判定した総数}}$$

$$\text{Semantic Labeled Recall (SLR)} = \frac{\text{述語語義同定正解数} + \text{意味役割付与正解数}}{\text{述語総数} + \text{意味役割を持つ項の数}}$$

$$\text{Semantic Labeled F1 (SLF)} = \frac{2 \times \text{SLP} \times \text{SLR}}{\text{SLP} + \text{SLR}}$$

$$\text{Macro F1 Score (MFS)} = \frac{\text{LAS} + \text{SLF}}{2}$$

7言語のうち、4言語に関しては交差の有る依存構造木を含んでいる(チェコ語:13.94%, 英語:3.74%, ドイツ語:25.79%, 日本語:0.91%)。一方、本システムで用いている依存構造解析アルゴリズムは交差には対応していない。したがって、交差を扱う枠組みを導入する必要性が出てくる。そこで、交差のある木を交差の無い木に変換する枠組みである Pseudo-Projective Transformation²⁵⁾ の適用を考えた。しかし、この手法を用いた場合、交差のある木から交差の無い木への変換時に、元の係り受け関係は変換後の依存構造木のラベルにエンコードされるため、結果的に非常に多くの依存構造ラベルが生成され、解析時間が大幅に遅くなる。結局、本実験では依存構造に交差がある場合についても、その存在は無視して解析をおこなった。

実験の結果を表5に示す。述語項構造解析において手法の変更をおこなったため、Official Results⁸⁾ とは結果が異なっている。“PR”は、品詞タガーと lemmatizer によって付与された品詞と見出し語を用いた結果であり、これは CoNLL-2009 Shared Task と同様の実験設定である。“CR”は正解の品詞と見出し語を用いた結果を示している。

この結果を CoNLL-2009 Shared Task における他の参加チームの結果と比較すると、本システムは英語で2位、チェコ語で3位という結果になっており、一部の言語ではあるが、トッ

表5 CoNLL-2009 Shared Task データでの依存構造および述語項構造の解析結果

		Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
PR	MFS	78.39	75.02	73.86	81.58	87.02	69.31	84.69	77.25
	SLF	75.96	71.42	75.02	85.06	85.50	63.81	77.67	73.27
	LSA	80.77	78.62	72.66	78.10	88.54	74.60	91.66	81.23
CR	MFS	84.93	85.22	82.03	83.25	88.62	83.70	85.33	84.40
	SLF	82.26	81.85	80.78	87.02	86.46	80.36	78.64	80.76
	LSA	87.02	88.59	83.27	79.48	90.77	87.03	91.96	88.04

プのチームと比較して遜色の無い結果が得られている。英語に関して、トップの成績を収めている文献 34) の手法との差が MFS で 0.67、チェコ語に関して、文献 3) の手法との差が MFS で 1.69 である。

6.1 依存構造解析の結果

チェコ語、英語、日本語については、品詞を正しいものからタガーで付与された品詞に置き換えても精度にそれほど差はみられないが、カタロニア語、中国語、ドイツ語、スペイン語では、正しい品詞からタガーで付与された品詞に置き換えたことで著しく精度が低下している(6.3 - 11.2%)。調査したところこれらの言語で用いられた品詞タガーの精度が比較的低い(92.3 - 95.5%) ことがわかった。これが依存構造解析の性能に大きく影響を及ぼしていると考えられる。

また、ドイツ語の依存構造解析の精度が悪くなっている他の理由としては、ドイツ語は交差を含む依存構造木の比率が比較的高いことがある。この問題に対処するためには、Pseudo-Projective Transformation をおこなうか、または最大全域木アルゴリズム²⁰⁾ などの交差を許容する依存構造解析アルゴリズムの適用が必要となる。

6.2 述語項構造解析の結果

大域的素性の有効性を検証するため、局所的素性のみで学習したモデル F_L と、局所的素性と大域的素性の双方を加えたモデル $F_L + F_G$ の比較実験をおこなった。この実験では、開発データでテストをおこない、正解の依存構造木を用いた。実験結果を表6に示す。この結果を見ると、全ての言語において大域的素性が有効に機能していることがわかる。精度と再現率を個別に評価した場合でも、日本語の精度を除き、全ての言語において、精度と再現率が向上している。

全7言語の述語項構造解析の結果のうち、日本語の結果が最も悪くなっている。これは、CoNLL-2009 Shared Task の日本語データの性質によるものと考えられる。このデータでは、文節単位である日本語の依存構造木が単語単位に変換されており、この変換には、文末以外

表 6 大域的素性によるパフォーマンスの変化。 F_L および F_L+F_G は SLF, ΔP および ΔR は F_L+F_G の結果と F_L only の精度, 再現率の差分を表している。

	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
F_L only	84.76	85.89	86.66	89.63	85.70	81.10	79.94	84.41
F_L+F_G	86.18	87.34	88.36	89.83	87.25	83.82	80.74	85.92
ΔP	+1.07	+1.46	+1.68	+0.21	+1.29	+2.54	-1.18	+1.51
ΔR	+1.69	+1.46	+1.70	+0.14	+1.77	+2.90	+2.37	+1.51

の単語は後続する次の単語に係けるという単純な規則が用いられている。意味役割の同定には、依存構造上のパスの情報は非常に重要な手がかりとなるが、そのような変換を用いた場合、述語と項の依存構造上での距離が長くなり、同定が難しくなる^{*1}。また、他の言語では語義とその語義がとる項の意味役割を定義したフレームが整備されており、その情報を用いる事が可能であるのに対し、日本語ではそのような情報は与えられていない。フレーム情報は意味役割同定において非常に重要な情報であるため、この情報の欠如が高精度な述語項構造解析を困難にしている。

7. 関連研究

依存構造解析と述語項構造解析はこれまで個別の問題として扱われてきたが、双方のタスクが相互に依存しながら最適な出力が得られるような結合学習手法への関心が高まってきており、これまでいくつかの手法が提案されている。

文献 13) では、二次交差なし依存構造解析の N-best 解に対して述語項構造解析をおこなう、最終的にリランキングをおこなうことにより依存構造と述語項構造の結合学習を実現している。この手法では、述語項構造解析において大域的素性を扱うためにリランキングを用いているが、本システムでは、大域的素性を単一のモデルで扱っているため手法が異なる。

文献 7), 10) では、依存構造と述語項構造の解析を、モードを切り替えながらおこなう手法が提案されている。この手法は依存構造解析に用いられる Shift-Reduce 法²³⁾ を述語項構造解析にも応用し、モードを切り替えるアクションを追加することにより、双方の解析が切り替わりながら最終的に双方のタスクの解析結果が得られるというものである。この手法は、アクションの選択は過去のアクションの結果に基づいて決定される。

文献 17), 18) では、Eisner アルゴリズムを拡張し、依存構造解析および述語項構造解析の

双方のスコアが全体として最大となるような構造を得る手法が提案されている。この手法は述語項構造解析に用いられる素性を得るために、事前に依存構造解析をしておく必要があるため純粋な結合学習手法とは言い難い。また、本システムと比較すると性能の面で課題が残っている⁸⁾。

一方、依存構造解析の後に述語項構造解析をおこなうパイプライン処理でも十分な性能が出ることは示されている。文献 3) では、既存の二次交差無し依存構造解析アルゴリズム^{2), 19)} で用いられている素性集合の拡張により、また、述語項構造解析においては、緻密な素性選択をおこなうことによりそれぞれのタスクについて高い解析精度を実現している。この手法では、二段階の依存構造解析における後段の処理では、整数計画法により必須の意味役割の重複を無にするなどの制約下での最適化をおこなっている。これは、本手法での大域的素性の役割と類似していると考えられるが、手法は異なる。また、文献 3) の方法では再現率の低下が報告されている一方、本手法ではほぼ全ての言語において精度、再現率双方の改善に成功しており、この点において本手法の優位性が認められる。

また文献 34) では、依存構造解析では単語クラスタリングの結果を依存構造解析の素性として使用する手法¹⁶⁾ や、複数の解析器を用意し、一方の解析器の結果をもう一方の解析器の素性として用いるスタッキング法²⁴⁾ を適用することにより、述語項構造解析では貪欲素性選択法 (Greedy Feature Selection) を用いて最適な素性集合を得ることにより、双方のタスクにおいて高い解析精度を実現している。しかしこの手法では、本手法で扱える大域的素性は用いられていない。

8. まとめ

本稿では、オンライン最大マージン学習アルゴリズムに基づく多言語依存構造解析および述語項構造解析システムについて述べた。本システムは依存構造解析と述語項構造解析は完全なパイプラインで処理が行われるシステムではあるが、CoNLL-2009 Shared Taskにおいて、一部の言語に関してはトップのチームと遜色の無い結果が得られた。また、依存構造解析において、局所的素性と大域的素性を単一のモデルにて扱うことが可能な、新たなオンライン最大マージン学習アルゴリズムを提案し、その有効性を示した。

今後の研究課題としては、まずは有効な大域的素性に関する調査が挙げられる。本稿の実験にて使用した大域的素性は僅か 2 種類であるため、未だに研究の余地が残されていると考えられる。また、効率的な依存構造および述語項構造の結合推論手法に関しても、未だ検討の余地がある。本稿で述べたシステムは、文献 13) で提案されている N-best 解に基づく

*1 述語と項の依存構造上の距離の平均は、英語で 1.38, チェコ語で 1.46, ドイツ語で 1.28 であるのに対し、日本語は 3.02 であった。

結合推論手法の適用が可能であるが、二次素性を用いた交差無し依存構造解析アルゴリズムは、 n を文長、 L を依存構造ラベルの種類数とすると、 argmax の計算に $\mathcal{O}(n^4 L)$ と比較的重い計算が必要である。依存構造木の N-best 解をビームサーチアルゴリズムなどの方法により得る場合はさらに多くの計算が必要となるが、これは実アプリケーションでの使用を考慮した場合に現実的な選択肢とは言い難い。効率的に相互のタスクの利点を生かせるような結合推論手法の枠組みについて、今後検討していきたいと考えている。

謝 辞

依存構造解析器の実装に関して有益なコメントをくださった Richard Johansson 氏に感謝いたします。また、データを提供して頂いた CoNLL-2009 Shared Task のオーガナイザーの皆様^{1),8),9),14),27),29),30)} に感謝いたします。

参 考 文 献

- 1) Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S. and Pinkal, M.: The SALSA corpus: a German corpus resource for lexical semantics, *LREC-2006*, Genoa, Italy (2006).
- 2) Carreras, X.: Experiments with a Higher-Order Projective Dependency Parser, *EMNLP-CoNLL 2007* (2007).
- 3) Che, W., Li, Z., Li, Y., Guo, Y., Qin, B. and Liu, T.: Multilingual Dependency-based Syntactic and Semantic Parsing, *CoNLL-2009* (2009).
- 4) Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S. and Singer, Y.: Online Passive-Aggressive Algorithms, *JMLR*, Vol.7, pp.551–585 (2006).
- 5) Daumé III, H.: Practical Structured Learning Techniques for Natural Language Processing, PhD Thesis, University of Southern California, Los Angeles, CA (2006).
- 6) Eisner, J.: Three New Probabilistic Models for Dependency Parsing, *ICCL 1996* (1996).
- 7) Gesmundo, A., Henderson, J., Merlo, P. and Titov, I.: A Latent Variable Model of Synchronous Syntactic-Semantic Parsing for Multiple Languages, *CoNLL-2009* (2009).
- 8) Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N. and Zhang, Y.: The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages, *CoNLL-2009*, Boulder, Colorado, USA (2009).
- 9) Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M. and Žabokrtský, Z.: Prague Dependency Treebank 2.0 (2006).
- 10) Henderson, J., Merlo, P., Musillo, G. and Titov, I.: A Latent Variable Model of Synchronous Parsing for Syntactic and Semantic Dependencies, *CoNLL 2008* (2008).
- 11) Johansson, R.: Dependency-based Semantic Analysis of Natural-language Text, PhD Thesis, Lund University (2008).
- 12) Johansson, R. and Nugues, P.: Extended Constituent-to-dependency Conversion for English, *NODAL-IDA 2007* (2007).
- 13) Johansson, R. and Nugues, P.: Dependency-based Syntactic-Semantic Analysis with PropBank and NomBank, *CoNLL 2008* (2008).

- 14) Kawahara, D., Kurohashi, S. and Hasida, K.: Construction of a Japanese Relevance-tagged Corpus, *LREC-2002*, Las Palmas, Canary Islands, pp.2008–2013 (2002).
- 15) Kazama, J. and Torisawa, K.: A New Perceptron Algorithm for Sequence Labeling with Non-local Features, *EMNLP-CoNLL 2007* (2007).
- 16) Koo, T., Carreras, X. and Collins, M.: Simple Semi-Supervised Dependency Parsing, *ACL 2008* (2008).
- 17) Lluís, X., Bott, S. and Màrquez, L.: A Second-Order Joint Eisner Model For Syntactic and Semantic Dependency Parsing, *CoNLL-2009* (2009).
- 18) Lluís, X. and Màrquez, L.: A Joint Model for Parsing Syntactic and Semantic Dependencies, *CoNLL 2008* (2008).
- 19) McDonald, R.: Discriminative Learning and Spanning Tree Algorithms for Dependency Parsing, PhD Thesis, University of Pennsylvania (2006).
- 20) McDonald, R., Pereira, F., Ribarov, K. and Hajič, J.: Non-projective Dependency Parsing using Spanning Tree Algorithms, *HLT-EMNLP 2005* (2005).
- 21) McDonald, R. and Pereira, F.: Online Learning of Approximate Dependency Parsing Algorithms, *EACL 2006* (2006).
- 22) Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B. and Grishman, R.: The NomBank Project: An Interim Report, *HLT-NAACL 2004 Workshop on Frontiers in Corpus Annotation* (2004).
- 23) Nivre, J., Hall, J., Nilsson, J., Eryigit, G. and Marinov, S.: Pseudo-Projective Dependency Parsing with Support vector Machines, *CoNLL 2006* (2006).
- 24) Nivre, J. and McDonald, R.: Integrating Graph-based and Transition-based Dependency Parsers, *ACL 2008* (2008).
- 25) Nivre, J. and Nilsson, J.: Pseudo-Projective Dependency Parsing, *ACL 2005* (2005).
- 26) Palmer, M., Gildea, D. and Kingsbury, P.: The Proposition Bank: An Annotated Corpus of Semantic Roles, *Computational Linguistics*, Vol.31, pp.71–106 (2005).
- 27) Palmer, M. and Xue, N.: Adding semantic roles to the Chinese Treebank, *Natural Language Engineering*, Vol.15, No.1, pp.143–172 (2009).
- 28) Shen, D. and Lapata, M.: Using Semantic Roles to Improve Question Answering, *EMNLP-CoNLL-2007* (2007).
- 29) Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L. and Nivre, J.: The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies, *CoNLL-2008* (2008).
- 30) Taulé, M., Martí, M.A. and Recasens, M.: AnCor: Multilevel Annotated Corpora for Catalan and Spanish, *LREC-2008*, Marrakesh, Morocco (2008).
- 31) Toutanova, K., Haghighi, A. and Manning, C.D.: Joint Learning Improves Semantic Role Labeling, *ACL 2005* (2005).
- 32) Watanabe, Y., Iwatate, M., Asahara, M. and Matsumoto, Y.: A Pipeline Approach for Syntactic and Semantic Dependency Parsing, *CoNLL 2008* (2008).
- 33) Wu, D. and Fung, P.: Can Semantic Role Labeling Improve SMT?, *EAMT-2009* (2009).
- 34) Zhao, H., Chen, W., Kazama, J., Uchimoto, K. and Torisawa, K.: Multilingual Dependency Learning: Exploiting Rich Features for Tagging Syntactic and Semantic Dependencies, *CoNLL-2009* (2009).
- 35) 飯田龍, 小町守, 乾健太郎, 松本裕治: NAIST テキストコーパス: 述語項構造と共参照関係のアノテーション, 情報処理学会自然言語処理研究会予稿集, NL-177-10 (2007).
- 36) 風間淳一, 鳥澤健太郎: 大域的素性を用いたタグ付けのためのパーセプトロン学習, 言語処理学会第 13 回年次大会 (2007).