情報処理学会論文誌　数理モデル化と応用　Vol.6　No.3　43–52 (Dec. 2013)

カテゴリ階層を考慮した構造化パーセプトロンによる固有表現抽出

東山 翔平[1,a)]　ブロンデル マチュー[1,†1,b)]　関 和広[1,c)]　上原 邦昭[1,d)]

受付日 2012年11月8日，再受付日 2013年1月10日 / 2013年3月12日，
採録日 2013年4月16日

概要：固有表現抽出は，テキスト中に現れる人名などの語句の同定を目的とする自然言語処理の基本的な問題である．抽出する固有表現は，人名や組織名など数種類を対象とすることが一般的であり，これらのカテゴリの間の関係は考慮しないことが多い．しかし，これらのカテゴリは階層性を有する場合があり，その場合，階層的に近い（遠い）という情報は抽出の際に活用できる可能性がある．本研究では，階層構造が定義された固有表現を対象に，階層的な近さの値を与えるコスト関数を定義する．機械学習手法である構造化パーセプトロンにコスト関数を導入し，カテゴリの階層性を考慮した固有表現抽出法を提案する．GENIA コーパスを用いて階層構造を持つ固有表現の抽出実験を行い，提案手法により，抽出の誤りの程度を小さくするとともに，正しい固有表現の抽出精度を高めることが可能になることを示す．

キーワード：固有表現抽出，コスト考慮型学習，構造化パーセプトロン，GENIA コーパス

Named Entity Recognition Exploiting Category Hierarchy Using Structured Perceptron

Shohei Higashiyama[1,a)]　Blondel Mathieu[1,†1,b)]　Kazuhiro Seki[1,c)]　Kuniaki Uehara[1,d)]

Received: November 8, 2012, Revised: January 10, 2013/March 12, 2013,
Accepted: April 16, 2013

Abstract: Named Entity Recognition (NER) is a fundamental natural language processing task concerned with the identification and classification of expressions into predefined categories (e.g., *person*, *organization*, *location*, etc). Existing NER systems usually target around ten categories and do not take into account category relations. However, it is often the case that categories naturally belong to some predefined hierarchy. When such is the case, the distance between categories in the hierarchy becomes a rich source of information which can be exploited and is intuitvely particularly useful when the categories are numerous. In this paper, we propose an NER system which can leverage category hierarchy information by introducing, in the structured perceptron framework, a cost function that penalizes more strongly category predictions which are far in the hierarchy from the correct category. We demonstrate the effectiveness of the proposed method through experiments on the GENIA biomedical text corpus, in particular in comparison to methods which do not take into account category hierarchy.

Keywords: named entity recognition, cost-sensitive learning, structured perceptron, GENIA corpus

1. はじめに

固有表現抽出は，テキスト中に現れる人名などの語句の同定を目的とする自然言語処理の基本的な問題である．抽出する固有表現は，人名，地名，組織名など数種類を対象とすることが一般的であり，これらのカテゴリの間の関係

[1] 神戸大学大学院システム情報学研究科
Graduate School of System Informatics, Kobe University, Kobe, Hyogo 657–0013, Japan
[†1] 現在，NTT コミュニケーション科学基礎研究所
Presently with NTT Communication Science Laboratories
[a)] higashiyama@ai.cs.kobe-u.ac.jp
[b)] mblondel@ai.cs.kobe-u.ac.jp
[c)] seki@cs.kobe-u.ac.jp
[d)] uehara@kobe-u.ac.jp

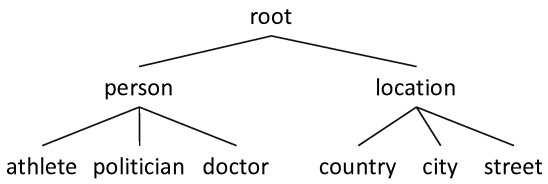


図 1　固有表現の階層の例
Fig. 1　An example hierarchy of named entities.

は考慮しないことが多い．しかし，これらのカテゴリは階層性を有する場合がある．たとえば，組織名はさらに企業名や大学名などのサブカテゴリに細分化されうる．

より詳細なサブカテゴリを持つ固有表現は，情報抽出などの自然言語処理タスクにおいて重要な役割を果たす．たとえば，テキスト情報を利用した株価予測の研究では，名詞が企業名であるかどうかの判断が必要になることがある．このような場合には，組織名の下に企業名というサブカテゴリがあった方が有益であり，一般に，幅広い情報を抽出できるようにするためには多様なカテゴリが必要となる．そのため，機械学習を用いて自動的に固有表現カテゴリを細分化する手法が提案されている [6], [7]．また，多数のサブカテゴリからなる階層構造を持つ固有表現カテゴリが提唱されている [19], [20]．

階層構造を持つ固有表現を対象とした固有表現抽出では，抽出の結果が単に正しいか誤っているかだけでなく，誤った場合でも，誤りの程度という尺度での評価を考えることができる．たとえば，**図 1** の階層構造を持つカテゴリの場合，単語 “Japan” にカテゴリ city，doctor を付与した場合，どちらも誤りである．しかし，city を付与した場合の方がより正解に近いと考えられる．固有表現抽出では，正しいラベルの付与を行い，抽出の精度を高めることが重要である．一方で，抽出された固有表現はより高次の自然言語処理タスクに影響を与えるため，抽出の誤りの程度を小さくすることも必要である．

このような階層構造を持つカテゴリにおいては，階層的に近い（遠い）という情報を利用することで，より誤りの程度が小さい固有表現抽出が実現できる可能性がある．また，階層的な近さの情報は，カテゴリ数が多い場合に特に有用であると考えられる．

本研究では，カテゴリ間の階層構造が定義された固有表現を対象に，2 つのカテゴリが互いに近い位置にある場合に小さい値をとる階層コスト関数を定義する．機械学習手法である構造化パーセプトロン [2] に階層コスト関数を導入することで，カテゴリの階層性を考慮した固有表現抽出を行い，提案手法の有効性を検証する．

本論文の構成は以下のとおりである．まず，2 章で固有表現抽出の概要と関連研究について述べ，3 章で構造化パーセプトロンを用いた系列ラベリングの方法を説明する．4 章では，誤分類のコストを考慮する学習の枠組みについて述べ，構造化パーセプトロンにおいてコストを考慮した学習を行う方法を説明する．また，階層構造を持つ固有表現を対象とする階層コスト関数を定義し，構造化パーセプトロンに階層コスト関数を導入した手法を提案する．5 章では，階層構造を持つ固有表現のデータセットを用いてコスト関数についての評価実験を行い，その結果を考察する．最後に，6 章で本論文のまとめと今後の課題を述べる．

2. 固有表現抽出

2.1 固有表現の種類とカテゴリの階層構造

固有表現抽出は，評価型のワークショップである MUC-6（Message Understanding Conference-6）[10] において共通タスクとして設定されたことで，広く研究が行われるようになった．この際に定義された固有表現は，3 種類の固有名詞（person，location，organization）と 4 種類の数値表現（date，time，money，percent）からなる 7 種類である．IREX（Information Retrieval and Extraction Exercise）[18] では固有物名（artifact，商品名や書籍のタイトル名など）が追加され，ACE（Automatic Content Extraction）[4] では GPE（Geographical and Political Entity，政治的機能を持つ地名），施設名（facility，建物の名前など）の 2 つが追加されている．一方，CoNLL-2003 shared task [23] は，特定の言語に依存しないことを意図した固有表現抽出のワークショップであり，そのデータセットは近年の研究で広く用いられている．このワークショップで定義された固有表現は，person，location，organization，miscellaneous の 4 種類である．このように，従来では抽出の対象とされる固有表現は多くても 10 種類程度である．そして，これらの固有表現カテゴリの間の関係は考慮されていない．

また，固有表現カテゴリの細分化が，情報検索や質問応答システムなどの自然言語処理タスクや，オントロジの自動構築に有用であるとして，固有表現をサブカテゴリに細分類する研究が行われている．Fleischman ら [6], [7] は，決定木などの機械学習手法を用いて location を country，city，street などのサブカテゴリに分類し，同様に person を athlete，politician，doctor などに細分類した．Sekine ら [19], [20] は，200 種類のカテゴリからなる拡張固有表現階層を定義し，日本語および英語のコーパスを構築している．拡張固有表現は，美術博物館名（museum），河川名（river）などの多数の詳細なサブカテゴリや，製品名（product），イベント名（event）などの広範なカテゴリを含む．Ohta ら [16] は，生物医学分野のタグ付きコーパスである GENIA コーパスを作成し，その際にタンパク質名や遺伝子名などの固有表現からなる階層構造を構築している．

2.2 固有表現抽出に用いられる手法

近年の固有表現抽出の研究では，機械学習，特に，教師

あり学習の手法が用いられるのが一般的である．

CoNLL-2003 shared task では，最大エントロピー法（Maximum Entropy Method）を利用したシステムが最もよく用いられた．特に，その英語のデータセット[*1]では，Florian ら [8] が最大エントロピー法に隠れ Markov モデルなどを組み合わせた手法で最も高い精度を達成し，Chieu ら [1] が最大エントロピー法を用いて 2 番目に高い精度を達成している．

最大エントロピー法を構造学習[*2]に拡張した手法として Conditional Random Fields（CRF）がある．Finkel ら [5] は，CRF において一般的に用いられる Viterbi や Forward-Backward などの推定アルゴリズムの代わりに，Gibbs サンプリングを用いて推定を行う手法を提案した．CoNLL-2003 shared task の英語のデータセットに適用し，同ワークショップの参加者に対して比較的高い精度であることを報告している．

Tsochantaridis ら [24] は，Support Vector Machine（SVM）を構造学習に拡張した構造化 SVM を提案した．固有表現抽出に適用し，構造化パーセプトロン [2] や CRF よりエラー率が低いことを報告している．

一方，階層構造を有する固有表現を対象とした研究としては，GENIA コーパスを使用したものがある．GENIA コーパスでは，分子生物学の専門用語 48 カテゴリからなる階層構造（木構造）が定義され，その根ノードに相当する 36 カテゴリがアノテーションに用いられている．ただし，GENIA コーパスを使用した研究には，一部のカテゴリのみを抽出の対象としたものが多い．たとえば，GENIA コーパスをデータセットに用いたワークショップである JNLPBA shared task [13] では，タスクを単純化するため，対象とするカテゴリを 36 カテゴリのうち 5 つに限定している．全カテゴリを抽出の対象としている研究としては，Lee らの研究 [14] がある．Lee らは，固有表現の認識，各固有表現のクラスへの分類という 2 段階の分類問題として固有表現抽出を定式化し，SVM を適用した．単純に分類を行った手法や，ルールベースによる方法より精度が良いことを示している．

2.3　固有表現抽出タスクの概要

固有表現抽出は，入力文中の固有表現部分を同定するタスクである．入力文中の各単語に対して，固有表現カテゴリとチャンクタグからなるタグをラベルとして付与することでこの部分を同定できる．カテゴリは人名（person），地名（location）といった固有表現の種類を表し，チャンクタグは特定のチャンク中における位置を表す．代表的なチャンクタグである IOB2 では，B，I，O の 3 種類のタグがあり，B が範囲の開始位置，I が範囲の内部，O が範囲の外側を表す．たとえば，単語列 “in New York City” 内の単語に，それぞれラベル O，B-LOCATION，I-LOCATION，I-LOCATION が付与されている場合，“New York City” が地名を表す 1 つの固有表現であることを表すことになる．

3.　構造化パーセプトロンによる系列ラベリング

3.1　系列ラベリングと固有表現抽出

トークン列 $\boldsymbol{x} = (x_1, \ldots, x_T)$ の各要素 x_t に適切なラベル y_t を付与する問題を系列ラベリング問題という．これは，トークン列 $\boldsymbol{x}$ に対してラベルの列 $\boldsymbol{y} = (y_1, \ldots, y_T)$ を付与する問題と考えることができる．

固有表現抽出は系列ラベリングの代表的なタスクの 1 つである．固有表現抽出では，トークン列 $\boldsymbol{x}$ は単語を要素とする文であり，ラベル y_t は固有表現タグ（B-LOCATION などチャンクタグとカテゴリの対からなるタグ）または非固有表現タグ（チャンクタグ O）である．

3.2　パーセプトロンの適用範囲の広がり

パーセプトロンは，1958 年に Rossenblatt [17] が考案した機械学習手法である．学習アルゴリズムが非常に単純である一方で，Freund ら [9] により，最近の機械学習手法である SVM に近い分類精度であることが示されている．

Collins [2] は，Freund らの投票型パーセプトロン（Voted Perceptron）[9] を拡張し，系列ラベリング問題に構造化パーセプトロン（Structured Perceptron）を適用した．品詞タグ付け，基本名詞句同定といったタスクにおいて，最大エントロピー法を精度で上回る結果を示している．

構造化パーセプトロンは構造化 SVM の特殊な場合に相当し [15]，正則化を導入せず，学習のイテレーション t に依存するステップサイズ η_t を $\eta_t = 1$ と簡略化した場合と等価である．なお，ステップサイズを解析的に求めるアルゴリズムとしては，Shalev-Shwartz らの手法 [21] がある．本研究では，実装が簡単であり，比較的高精度な手法であることから構造化パーセプトロンを使用した．

3.3　構造化パーセプトロンの学習アルゴリズム

構造化パーセプトロンにおける学習では，評価関数 f と重み $\boldsymbol{w}$ が主要な役割を果たす．評価関数 $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ は，トークン列 $\boldsymbol{x} \in \mathcal{X}$ とラベル列 $\boldsymbol{y} \in \mathcal{Y}$ に対して実数値を返す関数であり，重み $\boldsymbol{w}$ は学習すべきパラメータである．ここで，$\mathcal{X}$ は可能なトークン列全体の集合，$\mathcal{Y}$ は可能なラベル列全体の集合を表す．評価関数 f は，固定された次元 d を持つ 2 つのベクトル $\boldsymbol{w} \in \mathbb{R}^d$ と $\Phi(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ の内積で定義される．

[*1] CoNLL-2003 shared task では，英語とドイツ語のデータセットが用意されている．

[*2] 機械学習において，出力が構造を有する（出力中に依存関係が存在する）問題を構造学習という．3.1 節で後述する系列ラベリングは，構造学習の代表的な問題である．

$$f(\boldsymbol{x},\boldsymbol{y}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y}) \rangle \quad (1)$$

$\Phi(\boldsymbol{x},\boldsymbol{y})$ はトークン列とラベル列によって生成される素性ベクトルであり，素性ベクトルを返す関数 $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ は素性関数と呼ばれる．

評価関数 f は，トークン列 $\boldsymbol{x}$ に対するラベル列 $\boldsymbol{y}$ のスコア $f(\boldsymbol{x},\boldsymbol{y})$ を与えていると解釈できる．トークン列 $\boldsymbol{x}$ が与えられた際，可能なラベル列全体 $\boldsymbol{y} \in \mathcal{Y}$ において最も高いスコアをとる $\boldsymbol{y}$ を $\boldsymbol{x}$ の予測ラベル列 $\hat{\boldsymbol{y}}$ として出力する．

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \ f(\boldsymbol{x},\boldsymbol{y}) \quad (2)$$

構造化パーセプトロンでは，訓練事例 $\boldsymbol{x}^i$ の正解ラベル列 $\boldsymbol{y}^i$ に関する評価関数の値 $f(\boldsymbol{x}^i,\boldsymbol{y}^i)$ が，他のラベル列 $\boldsymbol{y} \neq \boldsymbol{y}^i$ の評価関数の値 $f(\boldsymbol{x}^i,\boldsymbol{y})$ より大きくなるように重み $\boldsymbol{w}$ を学習する．訓練データ中の事例に対してラベル列を推定し，推定されたラベル列 $\hat{\boldsymbol{y}}$ が正解ラベル列 $\boldsymbol{y}^i$ と異なる場合に，式 (3) により重みを更新することで逐次学習を行う．$\boldsymbol{w}^i$ は，i 番目の訓練事例を学習した後の重みを表している．

$$\boldsymbol{w}^i = \boldsymbol{w}^{i-1} + \Phi(\boldsymbol{x}^i,\boldsymbol{y}^i) - \Phi(\boldsymbol{x}^i,\hat{\boldsymbol{y}}^i) \quad (3)$$

トークン x に付与されうるすべてのラベルの集合を $\mathcal{L}$，その要素数を N とすると，長さ T のトークン列 $\boldsymbol{x}$ に対するラベル列の推定は，可能な N^T 個のラベル列 $\boldsymbol{y} \in \mathcal{L}^T$ $(= \mathcal{Y})$ の中から，評価関数 f の値を最大にするものを求める N^T クラスの多値分類問題と考えることができる．しかし，T の値が大きくなるとクラス数が膨大になりやすく，N^T 通りのラベル列に対する評価関数の値をすべて計算するのは現実的でない．そのため，動的計画法の一種である Viterbi アルゴリズムによりラベル列の推定を行う．

3.4　重みの平均化

Collins [2] は，重みの学習の最後に平均化（parameter averaging）という処理を行うことで，推定精度が向上することを示している．j 回目の学習において，i 番目の事例を学習し終えた後の重みを $\boldsymbol{w}^{j,i}$ と表記すると，平均化した重み $\bar{\boldsymbol{w}}$ は次式で定義される．ここで，m は学習回数，n は訓練事例の数である．

$$\bar{\boldsymbol{w}} = \frac{1}{nm} \sum_{j=1}^{m} \sum_{i=1}^{n} \boldsymbol{w}^{j,i}$$

m 回の学習を終えた後の重み $\boldsymbol{w}^{m,n}$ の代わりに，$\bar{\boldsymbol{w}}$ を用いてラベル列の推定を行うのが平均化の処理である．本研究でも，平均化した重みを用いて推定を行う．

4.　コスト考慮型学習

4.1　コスト考慮型学習とコスト関数

誤った分類に対するコストを考慮した機械学習の枠組みをコスト考慮型学習（Cost-Sensitive Learning）という．コスト考慮型学習では，単に分類の正解率を高めるのではなく，異なる誤りに異なる損失（コスト）を与え，平均的なコストが小さくなるように学習が行われる．

既存のコスト考慮型の学習アルゴリズムとしては，たとえば，Crammer らの Passive-Aggressive（PA）[3] がある．Crammer らは，マージンパーセプトロンの一種である PA を多値分類問題に適用して，通常のパーセプトロンよりもエラー率が低いことを示し，同時に，PA にコスト関数を加えたモデルも提案した．

コスト考慮型学習を実問題に適用した研究としては，Johansson らの研究 [11] がある．Johansson らは，意味役割付与のタスクにおいて，コスト考慮型の PA を利用して分類を行った．文の係り受け構造に注目した手法を提案し，意味役割付与において一般的であった構成素構造に基づくシステムと比較を行い，同等に近い精度であることを報告している．しかし，コスト関数を使用しない場合との比較については報告されておらず，コスト関数を用いたことがどの程度システムの性能に寄与したのかは明らかでない．また，Johansson らの手法では，入力文に対する解析木を学習する際などにコスト考慮型の PA を使用している．この際に用いられたコスト関数は，正解に対応する 0，完全な誤りに対応する 1 のほかに，中間的な誤りに対応する 0.5 という 3 つの値をとるという比較的単純なものである．

本研究では，カテゴリの階層性を考慮した誤りの程度の小さい固有表現抽出を実現するため，コスト関数を導入した構造化パーセプトロンにより抽出を行う．この際に用いるコスト関数として，カテゴリの階層構造に基づき，互いに近いカテゴリに対して小さく，遠いカテゴリに対して大きい値をとる階層コスト関数を提案する．5 章で述べる評価実験では，階層コスト関数のほかに，0 と 1 の値のみをとる最も単純なコスト関数を導入した構造化パーセプトロンおよび，コスト関数を使用しない通常の構造化パーセプトロンとの比較を行い，コスト関数の有効性について定量的な評価を行う．

4.2　コスト考慮型構造化パーセプトロン

分類問題におけるコスト関数は，ラベルの組を引数にとり，非負実数値を返す関数 $c : \mathcal{L} \times \mathcal{L} \to \mathbb{R}_+$ として定義できる（$\mathbb{R}_+$ は 0 以上の実数全体の集合を表す）．以後，これをラベル間のコスト関数と呼ぶ．コスト関数は，予測ラベルと正解ラベルを引数として，2 つのラベルが等しいときに 0，異なる場合に正の値をとることで誤分類のコストを与える．たとえば，具体的な定義としては，2 つのラベルが等しいか異なるかに応じて 0 または 1 をとるという最も単純なコスト関数（本論文ではこれを Hamming コスト関数と呼ぶ）が考えられる．

続いて，系列ラベリングにおけるコスト関数は，式 (4)

のように，系列中の要素に対するコストの和として定義できる．$\boldsymbol{y} = (y_1, \ldots, y_T)$，$\boldsymbol{y}' = (y'_1, \ldots, y'_T) \in \mathcal{Y}$ は，互いに長さが等しいラベル列とする．

$$C(\boldsymbol{y}, \boldsymbol{y}') = \sum_{t=1}^{T} c(y_t, y'_t) \tag{4}$$

以後，これをラベル列間のコスト関数と呼ぶ．

構造化パーセプトロンにおいてコストを考慮した推定を行うには，これらのコスト関数を用いて，式 (5) により学習を行うことで実現できる．

$$f_C(\boldsymbol{x}^i, \boldsymbol{y}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}^i, \boldsymbol{y}) \rangle + \alpha C(\boldsymbol{y}, \boldsymbol{y}^i) \tag{5}$$

通常の学習における評価関数 f と区別するため，コスト関数を導入した評価関数を f_C と表記した．式 (5) 中の α は正の実数値をとるパラメータであり，コスト関数が学習に与える影響を制御する役割を果たす．

正解ラベル列の情報を利用できる訓練の際は，評価関数 f_C によりラベル列 $\boldsymbol{y}$ のスコアを与え，次式のように f_C を用いて最大のスコアをとるラベル列を出力する．

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \ f_C(\boldsymbol{x}^i, \boldsymbol{y})$$

出力が正解と異なる場合の処理はコストを考慮しない場合と同様で，式 (3) により重みの修正を行う．一方，正解ラベル列が未知であるテスト事例に対しては，式 (1) の f を用いて予測ラベル列を推定する．

4.3　コスト関数の導入による学習への影響

前節において，構造化パーセプトロンでコストを考慮した推定を行う方法について述べた．本節では，コスト関数の導入が学習に与える影響について議論する．

コストを考慮した場合の学習について，式 (5) の評価関数 f_C によるラベル列のスコアに注目すると，元の「内積スコア」$\langle \boldsymbol{w}, \Phi(\boldsymbol{x}^i, \boldsymbol{y}) \rangle$ が，コスト関数の項 $C(\boldsymbol{y}, \boldsymbol{y}^i)$ により底上げされた状態になっている．つまり，ラベル列 $\boldsymbol{y}$ に対して，C の中で用いるラベル間のコスト関数 c の定義に基づいた正解ラベル列 $\boldsymbol{y}^i$ との距離が加算されている．正解から遠いラベル列ほど大きく底上げされているため，初めのうちは，そのような遠いラベル列が予測されることが多くなる．しかし，誤った予測が行われた際には，式 (3) により，誤った予測ラベル列の内積スコアが小さくなると同時に正解ラベル列の内積スコアが大きくなるように重み $\boldsymbol{w}$ が修正される．そのため，遠いラベル列のスコアは徐々に小さくなっていく．

また，互いに近いラベル列の素性ベクトルは共通する要素が多い傾向がある．内積スコアは素性ベクトルと重みとの内積で与えられるため，正解ラベル列の内積スコアを大きくすることは，正解に近いラベル列の内積スコアも共通の要素数に応じて大きくすることを意味する．正解から遠いラベル列の内積スコアを小さくする場合も同様である．したがって，正解ラベル列のスコアが十分大きくなるまで学習を繰り返すと，近いラベル列は近さに応じてスコアが大きくなり，遠いラベル列のスコアは遠いものほど小さくなる傾向が生じると考えられる（近いラベル列が誤って予測された場合は，正解との素性ベクトルの類似性により，式 (3) 右辺の素性ベクトルの多くの要素が打ち消された状態になり，重みの変化，すなわちスコアの変化にあまり影響を与えない）．

一方，式 (1) の f を用いたテストでは，コスト関数による不正解への底上げがないため，正解と不正解のスコアの差が学習時より大きくなっている．

このように，f_C による学習の特徴は，遠いラベル列を積極的に重みの修正に用いる点と，テストの際に不正解ラベル列のスコアの底上げがなくなることで，正解ラベル列のスコアが相対的に大きくなる点にある．これらの理由により，f による通常の学習に比べて，正解に近いラベル列の予測が増加し，遠いラベル列の推定が少なくなると考えられる．

特に，ラベル間のコスト関数として階層構造に基づくコストを与えるものを用いると，正解に近いラベル列は，正解ラベル列と一致する要素が多いだけでなく，異なる要素も正解ラベルに階層的に近いラベルであるものとなるため，誤分類についても誤りの程度が小さい推定が多くなると考えられる．

4.4　階層コスト関数の定義

コスト考慮型の構造化パーセプトロンで用いるラベル間のコスト関数として，カテゴリの階層性に基づくコストを与える階層コスト関数を定義する．

固有表現を構成する単語が異なるカテゴリの固有表現として誤分類されたときのコストは，階層構造におけるカテゴリ間の距離で与えることができる[*3]．しかし，固有表現抽出におけるラベルは，チャンクタグ O またはチャンクタグ B，I とカテゴリ名との対であり，単語に付与されたラベルをもとにコストを与えるには，ラベルどうしの距離を考える必要がある．

本研究では，階層構造は木構造であることを仮定し，カテゴリ間の距離としてグラフ上の距離を用いる．グラフ上の距離とは，グラフ内の一方のノードから他方のノードへの最短パスの長さで与えられる距離である．そして，カテゴリ間の距離を拡張することでラベル間の距離を与え，これをラベル間のコスト関数とする．

準備として，全カテゴリ数を K として，カテゴリ全体

[*3] ただし，本論文でいう距離とは，必ずしも数学における距離（metric）を意味しない．実際，後述するラベル間の距離は非退化性 $x = y \Leftrightarrow d(x, y) = 0$（$x$，$y$ は距離空間の点，d は距離関数）を満たさないため，厳密には擬距離となる．

の集合を $\mathcal{E} = \{e_1, \ldots, e_K\}$，固有表現タグ全体の集合を $\mathcal{L}_e = \{l_{1B}, l_{1I}, \ldots, l_{KB}, l_{KI}\}$，非固有表現タグを l_O とする．l_{kB}，l_{kI} がそれぞれチャンクタグ B，I とカテゴリ e_k との対に対応しており，l_O がチャンクタグ O に対応している．このとき，ラベル全体の集合は $\mathcal{L} = \mathcal{L}_e \cup \{l_O\}$ となる．

続いて，固有表現タグにカテゴリを対応させる写像 $\varphi : \mathcal{L}_e \to \mathcal{E}$ を次のように定義する．

$$\begin{aligned} \varphi(l_{kB}) &= e_k \\ \varphi(l_{kI}) &= e_k \end{aligned} \tag{6}$$

φ は，B, I が付加されたカテゴリ（B-PERSON, I-PERSON など）を付加される前のカテゴリ（PERSON など）と同一視する写像である．

そして，カテゴリ間の距離を与えるグラフ上の距離 $d : \mathcal{E} \times \mathcal{E} \to \mathbb{R}_+$ および写像 φ を用いて，ラベル l, $l' \in \mathcal{L}$ に対する階層コスト関数 $c_{\text{hie}} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}_+$ を式 (7) で定義する．

$$c_{\text{hie}}(l, l') = \begin{cases} d(\varphi(l), \varphi(l')) & (l, l' \in \mathcal{L}_e) \\ 0 & (l = l' = l_O) \\ M & (\text{otherwise}) \end{cases} \tag{7}$$

右辺の一番下の M は階層構造に依存する定数であり，階層コスト関数 c_{hie} における最大値である．l と l' の一方のみが l_O である場合がこれに相当する．本手法では，最大値 M として次式を用いた．右辺第 1 項は d のとりうる最大値である．

$$M = \max_{e, e' \in \mathcal{L}_e} \{d(e, e')\} + 1$$

このようにして定義したコスト関数 c_{hie} により，固有表現タグ，非固有表現タグを含むすべてのラベルについての互いの近さの値が与えられる．本手法では，式 (4) 中の c として c_{hie} を用いたコスト関数 C を評価関数 f_C に導入することで，階層構造に基づくコストを考慮した推定を行い，より誤りの程度の小さい固有表現抽出の実現を目指す．

5. 評価実験

5.1 実験データ

本手法の有効性を検証するため，階層構造を有する固有表現のデータセット GENIA コーパス v3.02 [*4] を用いて評価実験を行った．GENIA コーパスは，分子生物学の論文アブストラクトに専門用語情報などをタグ付けしたコーパスである．タンパク質名や遺伝子名などの固有表現からなる階層構造が定義されており，固有表現カテゴリの種類は 36，階層の最大の深さは 7 である [*5]．

GENIA コーパスでは，1 つの単語（列）に複数の固有表現カテゴリが付与され，単語に付与される固有表現の間にネストが存在していることがある．本研究では，最もネストの浅い固有表現を抽出の対象とした．たとえば，“IL-2 gene expression and NF-kappa B activation through CD28 requires reactive oxygen production by 5-lipoxygenase.” という文では，“IL-2 gene expression” にタグ other_name が付与されると同時に，“IL-2 gene” にタグ DNA_domain_or_region が付与されている．この場合には，単語列 “IL-2 gene expression” は固有表現 other_name であるとした．

5.2 使用する素性

分子生物学分野の固有表現抽出で用いられる素性の多くが一般の固有表現抽出と共通している一方，ドメインに特有のものとして，ローマ数字，ギリシャ文字，A，T，C，G からなる塩基配列などがある．

そこで，本手法では，一般的な素性およびドメインに特有の素性として，単語自身と周辺単語，単語の部分文字列，上述の塩基配列などを含む単語の文字種を使用することにした．

周辺単語は注目している単語の前後 2 単語ずつとし，単語の部分文字列としては，訓練データ中で 30 回以上出現した接頭辞および接尾辞を用いた．文字種は，JNLPBA shared task で最も高性能であった Zhou らの手法 [25] が用いている文献 [22] の素性を用いた．この文字種に関する素性は，塩基配列やギリシャ文字などを含む 17 種類からなる [*6]．

これらの素性を，ラベル列 $\boldsymbol{y}$ 中の現在の位置 t におけるラベル y_t および，ラベル列中の連続する位置 $t-1$，t にあるラベルの組 (y_{t-1}, y_t) とともに用いた．$\mathcal{O}$ を訓練データ中の全単語の集合とすると，たとえば，単語 $o_k \in \mathcal{O}$ およびラベル l_i, $l_j \in \mathcal{L}$ に関する素性は，x_t を現在のトークンとして次のようになる．

$$\phi^0_{k,i}(x_t, y_t) = \begin{cases} 1 & (x_t = o_k, y_t = l_i) \\ 0 & (\text{otherwise}) \end{cases}$$

$$\phi_{k,i,j}(x_t, y_t, y_{t-1}) = \begin{cases} 1 & (x_t = o_k, y_t = l_i, y_{t-1} = l_j) \\ 0 & (\text{otherwise}) \end{cases}$$

5.3 評価方法

評価は，GENIA コーパスで訓練 9，テスト 1 の比率で 10 分割交差検定を行い，10 回の平均コストの平均を評価値に用いた．平均コストとは，テスト事例中の全トークンに対する階層コスト関数 c_{hie} の値の和を全事例について足

[*4] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/corpus/GENIAcorpus3.02.tgz から入手可能．

[*5] 文献 [12] に GENIA コーパス v3.0 の階層構造が図示されている．一部のカテゴリ名が異なるのを除いて，v3.02 も同じ階層構造を有する．

[*6] この 17 種類とは，文献 [22] における Simple deterministic features 22 種類のうち，“.”，“)” などの記号単体に関する素性を除いたものである．本手法では，これら記号単体の素性は単語の素性に含まれるため，文字種からは除外している．

し合わせ，出現単語数で割って正規化した値である．つまり，全テスト事例の集合を $\mathcal{T}$ とし，事例 $\boldsymbol{x} \in \mathcal{T}$ の要素 x_t に対する正解ラベルを y^t，式 (2) による予測ラベル列を $\hat{y}_t$ とすると，テストデータ $\mathcal{T}$ の平均コスト $\mathrm{cost}(\mathcal{T})$ は次式で表される．

$$\mathrm{cost}(\mathcal{T}) = \frac{1}{Z(\mathcal{T})} \sum_{\boldsymbol{x} \in \mathcal{T}} \sum_{x_t \in \boldsymbol{x}} c_{\mathrm{hie}}(\hat{y}_t, y_t)$$

ただし，$Z(\mathcal{T})$ は $\mathcal{T}$ に現れる単語の総数である．

平均コストは，テストデータ中の全単語に対する階層コスト関数の値の平均であり，値が小さいほど精度は良い．単語に対するコスト関数の値は，単語の予測ラベルが正解ラベルと等しいときに 0 となり，正解ラベルと異なるときは正解から遠いほど値が大きくなる．つまり，階層構造に基づいて誤りの程度を考慮していると考えることができ，この意味で，平均コストは階層性を考慮した評価尺度となっている．階層コスト関数の値の範囲はカテゴリの階層構造に依存し，実験に使用した GENIA コーパスの階層構造では，最小で 0，（非固有表現タグ O を含めて）最大で 11 である．

5.4　パラメータ α の影響

コスト関数を用いる際，式 (5) におけるパラメータ α の最適な値を決定する必要がある．そのため，以下のようにして α の値の決定を行った．

(1) 交差検定の各フォールドにおいて，訓練データをさらに学習用データと検定用データに分割する．データの分割の仕方としては，訓練データから事例をランダムに抽出し，75%を学習用データ，25%を検定用データとする．
(2) ある α の値に対し，各フォールドでは，学習用データで学習し，検定用データで平均コストを算出する．これを全フォールドについて平均し，その α における検定用データの精度とする．
(3) α の値を変化させながら上の手順 (2) を繰り返し，精度が最も高いときの α の値を最適値とする．

次節以降の実験では，このようにして決定した α の最適値を用いて交差検定を行った．交差検定の各フォールドでは，検定用データを含む元の訓練データで学習し，テストデータで精度を算出した．テストデータでの精度を全フォールドについて平均した値が最終的な精度である．

重みの学習における α の値の影響を調べるため，検定用データに対する精度の平均をプロットした．$0 \leq \alpha \leq 10$ の範囲で 0.2 刻みで α の値を変化させ，学習回数は評価値が収束するまでの 5 回とした．$\alpha = 0$ はコスト関数を使用しない場合と等価である．結果を**図 2** に示す．α の値が小さいときは，コスト関数が学習に与える影響が小さく，平均コストは大きい．α の値が大きくなると平均コストが減少する傾向があり，コスト関数が重みの学習に対して有効に作用していることが分かる．しかし，α の値が大きくなるにつれて減少は徐々に緩やかになり，$\alpha = 6$ を超えたあたりでほぼ横ばいになっている．そこで，$6 \leq \alpha \leq 10$ の範囲で，0.1 刻みの α の値について再び検定用データの精度を算出し，最も低い平均コストであった $\alpha = 9.0$ を最適値とした．

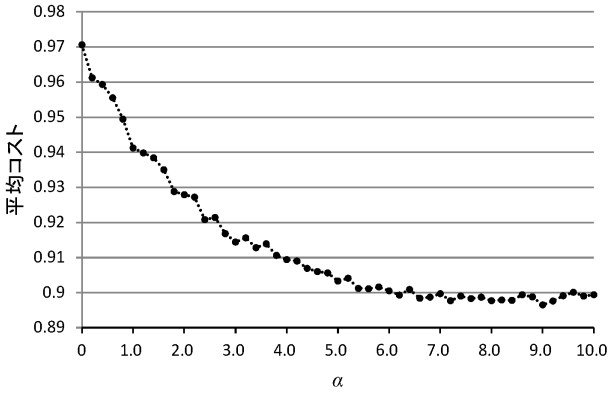


図 2　階層コスト関数（$0 \leq \alpha \leq 10$）における平均コスト
Fig. 2　Average cost when using the c_{hie} cost function ($0 \leq \alpha \leq 10$).

表 1　コスト関数の比較
Table 1　Comparison of cost functions.

	平均コスト	学習回数
ベースライン	0.9413	6
Hamming コスト関数（$\alpha = 47$）	0.9057	9
階層コスト関数（$\alpha = 9.0$）	**0.8823**	10

5.5　コスト関数の違いによる精度の比較

4.4 節では，グラフ上の距離を用いてラベル間の距離を与え，階層性を考慮する階層コスト関数 c_{hie} を定義した．しかし，コスト関数の定義は他にも考えられる．特に，カテゴリの階層的な情報を利用しないコスト関数は，階層性の考慮がどの程度有効に働いているかを検証する際に重要となる．そこで，2 つのラベルが表すカテゴリが等しければ 0，異なっていれば 1 をとる Hamming コスト関数 c_{ham} を次式で定義する[*7]．

$$c_{\mathrm{ham}}(l, l') = \begin{cases} 0 & (\varphi(l) = \varphi(l')) \\ 1 & (\mathrm{otherwise}) \end{cases} \tag{8}$$

本実験では，ベースラインとしてコスト関数を用いない通常の構造化パーセプトロン，階層コスト関数 c_{hie} を導入した場合，Hamming コスト関数 c_{ham} を導入した場合の 3 つの場合について，10 分割交差検定を行い，平均コストを算出した．Hamming コスト関数における最適な α の値は，5.4 節の階層コスト関数と同様に決定した．実験結果を**表 1** に示す．

[*7] 記述を簡潔にするため，式 (8) の φ は式 (6) の φ の始集合および終集合に l_O を含めたものとし，$\varphi(l_O) = l_O$ とする．

いずれのコスト関数を用いた場合でもベースラインより向上しており，特に，階層コスト関数を用いた場合に最も良くなっている．また，上記3つの場合の任意の2つの組合せについて，それぞれ一標本 t 検定を行ったところ，すべての検定において有意水準1%で有意差があった．この結果から，コスト関数を学習に用いることで誤りの程度が減少することが分かる．さらに，コスト関数を用いる場合は，階層性を考慮したものを用いることで，より誤りの程度が小さくなると結論できる．

一方，収束するまでの学習回数はコスト関数の有無によって異なり，コスト関数を使用した場合に回数が多くなっていることが分かる．現状では，学習の終了条件を評価値が収束するまでとしており，重みそのもの収束性や，分類誤差が一定以下になるまでに要する学習回数についての理論的な分析は行っていない．構造化パーセプトロンの分類誤差および重みの収束性は，文献 [2] の中で証明が述べられており，コスト関数を導入した場合についても，今後，同様の分析を行っていくことが必要である．

5.6　他のコスト考慮型学習手法との比較

コスト考慮型学習の他の手法として，Passive-Aggressive（PA）[3] との比較を行った．重みの更新には max-loss update を使用し，PA における正則化のパラメータ C_r [*8]は，5.4 節で本手法のパラメータ α を決定したのと同様に検定用データを用いて決定した．素性は 5.2 節と同様とし，階層コスト関数を用いた場合を対象に構造化パーセプトロンとの比較を行った．

PA は，マージンを確保するようにパラメータの学習を行うパーセプトロンの一種であり，学習アルゴリズムは構造化パーセプトロンと類似している．しかし，重みの更新の際のステップサイズ τ_i が一定ではなく，次式のように，出力ラベル列の正解からの遠さに依存する点で異なっている．

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \ \boldsymbol{w}\cdot\Phi(\boldsymbol{x}^i,\boldsymbol{y}) + C(\boldsymbol{y},\boldsymbol{y}^i)$$

$$\tau_i = \min\left(C_r, \frac{\boldsymbol{w}\cdot\Phi(\boldsymbol{x}^i,\hat{\boldsymbol{y}}) - \boldsymbol{w}\cdot\Phi(\boldsymbol{x}^i,\boldsymbol{y}^i) + C(\hat{\boldsymbol{y}},\boldsymbol{y}^i)}{\|\Phi(\boldsymbol{x}^i,\boldsymbol{y}^i) - \Phi(\boldsymbol{x}^i,\hat{\boldsymbol{y}})\|^2}\right)$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \tau_i(\Phi(\boldsymbol{x}^i,\boldsymbol{y}^i) - \Phi(\boldsymbol{x}^i,\hat{\boldsymbol{y}}))$$

検定用データを用いた予備実験では，パラメータ C_r の値を小さくすると，収束に要する学習回数が多くなる一方で，収束時の平均コストは小さくなる傾向があり，訓練時間と精度のトレードオフの関係にあった．C_r の各値に対して評価値が収束するまで学習を繰り返し，収束時の精度が最も高かった $C_r = 0.03$ を用いた．

訓練データ全体で再学習を行い，テストデータで精度を算出した結果を**表 2** に示す．参考のため，構造化パーセプトロン（SP）における精度および学習回数も再掲した．結果としては，平均コストは PA の方が低く，精度では構造化パーセプトロンがやや劣っている．ただし，2 つの手法の平均コストの差についての二標本 t 検定では，有意水準5%で有意差はなかった．また，PA では高い精度を得るまでに多くの学習回数を必要とするため，訓練時の計算コスト（学習回数）は構造化パーセプトロンの方が約 3 分の 1 と小さい．したがって，構造化パーセプトロンは，少ない訓練時間で PA と大きく変わらない精度が得られる手法であるといえる．

[*8] 文献 [3] では単に C と表記されているパラメータである．本論文では，ラベル列間のコスト関数 C と区別するため，C_r という表記を使用した．

表 2　Passive-Aggressive との比較
Table 2　Comparison with Passive-Aggressive.

	平均コスト	学習回数
SP+階層コスト関数（$\alpha = 9.0$）	0.8823	10
PA+階層コスト関数（$C_r = 0.03$）	**0.8789**	29

表 3　関連研究との比較
Table 3　Comparison with exsisting methods.

	F 値
構造化パーセプトロン	69.8
構造化パーセプトロン+階層コスト関数（$\alpha = 9.0$）	**70.7**
Lee et al., 2004	66.7

5.7　関連研究との比較

GENIA コーパスの固有表現全体を抽出の対象とている関連研究として，Lee らの手法 [14] を対象に本研究との比較を行った．

Lee らの手法では，固有表現抽出を単語を複数のクラスに分類する分類問題として定式化している．SVM を用いて，各単語が固有表現であるか否かに分類した後，固有表現のクラスに分類された単語を各固有表現カテゴリに分類している．GENIA コーパス v3.0p を用いて 10 分割交差検定を行い，全カテゴリに対する精度を F 値で 66.7 と報告している．

本手法のベースラインである構造化パーセプトロンおよび，階層コスト関数を導入した提案手法を用いて，10 分割交差検定における F 値（10 回のフォールドにおける F 値の平均）をそれぞれ算出した．パラメータ α の値および学習回数は前節と同様である．結果を**表 3** に示す．

本提案手法が 4 ポイント程度上回っており，コスト関数を用いないベースラインでも 3 ポイント程度上回っている．比較した手法では，各単語のラベルを別々に推定しており，1 つ前の単語のラベルが次の単語のラベルの推定に影響を与えない．一方，本研究で用いた構造化パーセプトロンでは，単語が直前の単語に依存するとして固有表現抽出を定式化しており，ある単語のラベルを推定する際には，それより前にある単語のラベルに影響を受ける．これは，

固有表現抽出のように，トークン（同じ文中の単語）間に依存関係があると考えられるタスクでは，トークン列全体に対して最適なラベル列を決定する手法がより有効であるためだと思われる．

ベースラインと提案手法の F 値に注目すると，階層コスト関数を用いることで F 値が向上している．こちらも平均コストと同様，一標本 t 検定において有意水準 1%で有意差があった．この結果から，コスト関数の導入により，誤りの程度が減少しただけでなく，正しい固有表現の抽出割合も増加していることが分かる．

6. おわりに

本研究では，カテゴリの階層構造が定義された固有表現を対象とした場合に，階層についての情報を有効に利用できると考え，構造化パーセプトロンに階層コスト関数を導入することで，階層性を考慮した固有表現抽出法を提案した．階層構造を持つ固有表現の抽出実験を行い，本手法により，抽出の誤りの程度を小さくするとともに，正しい固有表現の抽出精度を高めることが可能になることを示した．実験結果からは，構造化パーセプトロンの学習にコスト関数を導入することで，データに対して識別能力の高いパラメータの学習が可能になったと考えられる．

誤りの程度が減少したことは，言い換えれば，推定結果が誤りであっても正解に近い推定が多くなったということである．このことから，階層に基づくコストが小さくなるように学習を行うことで，誤分類であっても，推定されたカテゴリのトップカテゴリ（ルートの 1 つ下のカテゴリ）が正解のそれと一致しているケースは増加していると考えられる．

本研究では，階層構造があらかじめ定義されているデータセットのみを対象とした．しかし，階層構造が定義されていないデータセットについても，トップより下のカテゴリを自動的に生成することで，階層性を考慮した抽出法を適用できる．そのようにして抽出されたトップより下のカテゴリをそのトップカテゴリと同一視することで，トップカテゴリの認識を行った場合，上述したトップカテゴリの一致性についての考察から，通常の抽出よりもトップカテゴリの認識精度が向上する可能性があると考えられる．そこで，階層構造を有する他の固有表現抽出のデータセットに適用するとともに，階層構造があらかじめ定義されていないデータセットについても階層性を考慮した抽出を行い，本手法の有効性をさらに検証していきたい．

謝辞　本研究は JSPS 科研費 25330363 の助成を受けたものです．

参考文献

[1] Chieu, H. and Ng, H.: Named Entity Recognition with a Maximum Entropy Approach, *Proc. 7th Conference on Natural Language Learning* (*CoNLL-2003*), pp.160–163 (2003).

[2] Collins, M.: Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms, *Proc. 2002 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp.1–8 (2002).

[3] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S. and Singer, Y.: Online Passive-aggressive Algorithms, *The Journal of Machine Learning Research* (*JMLR*), Vol.7, pp.551–585 (2006).

[4] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R.: The Automatic Content Extraction (ACE) Program — Tasks, Data, and Evaluation, *Proc. 4th International Conference on Language Resources and Evaluation* (*LREC*), pp.837–840 (2004).

[5] Finkel, J., Grenager, T. and Manning, C.: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling, *Proc. 43rd Annual Meeting on Association for Computational Linguistics* (*ACL*), pp.363–370 (2005).

[6] Fleischman, M.: Automated Subcategorization of Named Entities, *Proc. 39th Annual Meeting on Association for Computational Linguistics* (*ACL*), pp.25–30 (2001).

[7] Fleischman, M. and Hovy, E.: Fine Grained Classification of Named Entities, *Proc. 19th International Conference on Computational Linguistics* (*COLING*), pp.1–7 (2002).

[8] Florian, R., Ittycheriah, A., Jing, H. and Zhang, T.: Named Entity Recognition through Classifier Combination, *Proc. 7th Conference on Natural Language Learning* (*CoNLL-2003*), pp.168–171 (2003).

[9] Freund, Y. and Schapire, R.: Large Margin Classification Using the Perceptron Algorithm, *Machine Learning*, Vol.37, No.3, pp.277–296 (1999).

[10] Grishman, R. and Sundheim, B.: Message Understanding Conference-6: A Brief History, *Proc. 16th International Conference on Computational Linguistics* (*COLING*), pp.466–471 (1996).

[11] Johansson, R. and Nugues, P.: Dependency-based Semantic Role Labeling of PropBank, *Proc. 2008 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp.69–78 (2008).

[12] Kim, J., Ohta, T., Tateisi, Y. and Tsujii, J.: GENIA Corpus—A Semantically Annotated Corpus for Biotextmining, *Bioinformatics*, Vol.19, Suppl.1, pp.i180–i182 (2003).

[13] Kim, J., Ohta, T., Tsuruoka, Y., Tateisi, Y. and Collier, N.: Introduction to the Bio-entity Recognition Task at JNLPBA, *Proc. International Joint Workshop on Natural Language Processing in Biomedicine and its Applications* (*JNLPBA*), pp.70–75 (2004).

[14] Lee, K., Hwang, Y., Kim, S. and Rim, H.: Biomedical Named Entity Recognition Using Two-phase Model Based on SVMs, *Journal of Biomedical Informatics*, Vol.37, No.6, pp.436–447 (2004).

[15] Martins, A.: The Geometry of Constrained Structured Prediction: Applications to Inference and Learning of Natural Language Syntax, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, USA (2012).

[16] Ohta, T., Tateisi, Y. and Kim, J.: The Genia corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain, *Proc. 2nd International Conference on*

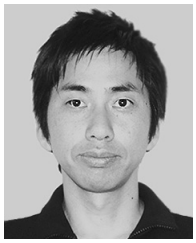
Human Language Technology Research (*HLT*), pp.82–86 (2002).
[17] Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, Vol.65, No.6, pp.386–408 (1958).
[18] Sekine, S. and Isahara, H.: IREX: IR and IE Evaluation Project in Japanese, *Proc. 2nd International Conference on Language Resources and Evaluation* (*LREC 2000*), pp.1475–1480 (2000).
[19] Sekine, S. and Nobata, C.: Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy, *Proc. 4th International Conference on Language Resources and Evaluation* (*LREC*), pp.1977–1980 (2004).
[20] Sekine, S., Sudo, K. and Nobata, C.: Extended Named Entity Hierarchy, *Proc. 3rd International Conference on Language Resources and Evaluation* (*LREC*), pp.1818–1824 (2002).
[21] Sharev-Shwaltz, S., Singer, Y. and Srebro, N.: Pegasos: Primal Estimated Sub-gradient Solver for Svm, *Proc. 24th International Conference on Machine Learning* (*ICML*), pp.807–814 (2007).
[22] Shen, D., Zhang, J., Zhou, G., Su, J. and Tan, C.: Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain, *Proc. ACL 2003 Workshop on Natural Language Processing in Biomedicine*, Vol.13, pp.49–56 (2003).
[23] Tjong Kim Sang, E.F. and De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition, *Proc. 7th Conference on Natural Language Learning* (*CoNLL-2003*), pp.142–147 (2003).
[24] Tsochantaridis, I., Hofmann, T., Joachims, T. and Altun, Y.: Support Vector Machine Learning for Interdependent and Structured Output Spaces, *Proc. 21st International Conference on Machine Learning* (*ICML*) (2004).
[25] Zhou, G. and Su, J.: Exploring Deep Knowledge Resources in Biomedical Name Recognition, *Proc. International Joint Workshop on Natural Language Processing in Biomedicine and its Applications* (*JNLPBA*), pp.96–99 (2004).

東山 翔平　（学生会員）

2012 年神戸大学工学部情報知能工学科卒業．現在，同大学院修士課程に在籍．情報抽出，固有表現抽出の研究に従事．言語処理学会会員．

ブロンデル マチュー

2008 年リール第 1 大学テレコム研究科修士課程修了．2012 年神戸大学大学院システム情報学研究科博士課程修了．現在，NTT コミュニケーション科学基礎研究所に在籍．博士（工学）．機械学習とその応用の研究に従事．



関 和広　（正会員）

2002 年図書館情報大学情報メディア研究科修士課程修了．2006 年インディアナ大学図書館情報学研究科博士課程修了．神戸大学助手，助教，講師を経て，現在，同大学院システム情報学研究科准教授．Ph.D. 情報検索，自然言語処理，機械学習の研究に従事．言語処理学会，ACM SIGIR 各会員．

上原 邦昭　（正会員）

1978 年大阪大学基礎工学部情報工学科卒業．1983 年同大学院博士後期課程単位取得退学．同産業科学研究所助手，講師，神戸大学工学部情報知能工学科助教授，同都市安全研究センター教授等を経て，現在，同大学院システム情報学研究科教授．工学博士．人工知能，特に機械学習，マルチメディア処理の研究に従事．電子情報通信学会，計量国語学会，日本ソフトウェア科学会，AAAI 各会員．