Vol.2012-MPS-91 No.25
Vol.2012-BIO-32 No.25
2012/12/7

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]　ブロンデル マチュー[1]　関 和広[1]　上原 邦昭[1]

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 10 種類程度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，抽出の精度を高めることが可能になることを示した．

Named Entity Recognition Exploiting Category Hierarchy Using Structured Perceptron

Higashiyama Shohei[1]　Blondel Mathieu[1]　Seki Kazuhiro[1]　Uehara Kuniaki[1]

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 a 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.

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，これらカテゴリ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．

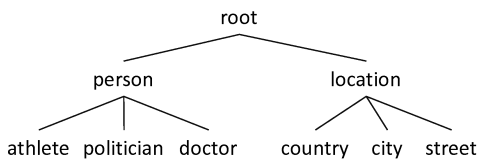

図 1　固有表現の階層の例
Fig. 1　An example hierarchy of named entities.

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，

[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

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，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，正しいラベルの付与を行うことだけでなく，抽出の誤りの程度を小さくすることも重要である．

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．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] にこの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．

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 [9] 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 [15] や ACE [3] といったワークショップでは，これに加え 1，2 の固有表現が追加されている．一方，CoNLL-2003 shared task [18] 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，オントロジー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 [5], [6] は，決定木を用いて 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．関根ら [16], [17] 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．大田ら [13] 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 [7] 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．

Finkel ら [4] は，最大エントロピー法を構造学習に拡張した CRF において，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 [19] は，SVM を構造学習に拡張した構造化 SVM を固有表現抽出に適用し，構造化パーセプトロン [2] や CRF よりエラー率で良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．ただし，GENIA コーパスを使用した研究には，36 あるカテゴリのうち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 JLNPBA shared task [11] 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 [12] 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問題である．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の対からなるタグをラベルとして付与する．カテゴリは人名，地名などの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，それぞれ範囲の開始位置，範囲の内部，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している．

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

[*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．

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 1 つの文であり，系列中の要素 x_t は単語である．また，ラベル y_t はカテゴリとチャンクタグの対からなるタグである．

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[14] 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 [8] 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 [8] を拡張し，系列ラベリング問題に構造化パーセプトロンを適用した．品詞タグ付けなどのタスクにおいて，最大エントロピー法を精度で上回る結果を示している．

3.3 構造化パーセプトロンの学習アルゴリズム

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 $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ を評価関数と呼ぶ．$\mathcal{X}$ 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である．評価関数 f は，等しい次元 d を持つ 2 つのベクトル，重み $\boldsymbol{w} \in \mathbb{R}^d$ と，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 $\Phi(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ の内積で定義される．

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

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」$f(\boldsymbol{x}, \boldsymbol{y})$ 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．

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

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．

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

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．この問題を解消するため，動的計画法の一種である Viterbi アルゴリズムを用いて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．m は学習回数，n は訓練事例数である．

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

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. 提案手法

4.1 評価関数へのコスト関数の導入

ラベルの組を引数にとり，非負実数値を返すコスト関数 $c : \mathcal{L} \times \mathcal{L} \to \mathbb{R}_+$ を考える．c は，2 つのラベル $l_i, l_j \in \mathcal{L}$ が階層構造において互いに近い位置にある場合に小さい値をとり，2 つのラベルが等しいときに値 0 をとるものとする．c の具体的な定義は後述する．

続いて，ラベル間のコスト関数 c を用いて，長さが等しい 2 つのラベル列 $\boldsymbol{y} = (y_1, \ldots, y_T), \boldsymbol{y}' = (y'_1, \ldots, y'_T) \in \mathcal{Y}$ について，ラベル列間のコスト関数 $C : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ を式 (4) で定義する．

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

3.3 節では，式 (1) の評価関数 f により，トークン列に対するラベル列のスコアを与えた．ところで，正解ラベル列が既知である訓練事例 $\boldsymbol{x}^i$ のラベル列の推定においては，各ラベル列 $\boldsymbol{y} \in \mathcal{Y}$ と正解ラベル列 $\boldsymbol{y}^i$ との近さの情報が利用可能である．そこで，f の代わりに，新しい評価関数 $f_C : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ を用いてラベル列 $\boldsymbol{y}$ のスコアを与えることを考える．f_C を式 (5) で定義する．

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

α はコスト関数が学習に与える影響を制御するパラメータであり，正の実数値をとる．

訓練データを用いた学習では，次式のように，f_C を用いて最大のスコアをとる予測ラベルを推定し，式 (3) により適宜重みの修正を行う．

$$\begin{aligned} \hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \ f_C(\boldsymbol{x}^i, \boldsymbol{y}) \\ &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \ \langle \boldsymbol{w}, \Phi(\boldsymbol{x}^i, \boldsymbol{y}) \rangle + \alpha C(\boldsymbol{y}^i, \boldsymbol{y}) \end{aligned}$$

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 を用いて推定を行う．

式 (5) の f_C によるラベル列のスコアは，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 $C(\boldsymbol{y}^i, \boldsymbol{y})$ により底上げされた状態になっている．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，始めのうちは，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．ラベル列の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，誤りの程度が大きい予測が少なくなると考えられる．

4.2 コスト関数の定義

2.3 節で述べたように，固有表現抽出におけるラベルはカテゴリとチャンクタグからなる．いま，固有表現カテゴリ全体の集合を $\mathcal{E} = \{e_1, \ldots, e_K\}$，固有表現タグ（B-PERSON 等）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（チャンクタグ O）を l_O とする．ただし，l_{kB}, l_{kI} がカテゴリ e_k とチャンクタグ B，I の対に対応している．このとき，ラベル全体の集合は $\mathcal{L} = \mathcal{L}_e \cup \{l_O\}$ となる．

固有表現カテゴリ，すなわち，$\mathcal{E}$ の元については，予め定義されている階層構造を利用することで，互いの近さを表す値が与えられる．本研究では，階層構造は木構造を仮定し，カテゴリ間の距離としてグラフ上の距離を用いる．つまり，階層構造中の一方のノードから他方のノードへの最短パスの長さを対応する 2 つのカテゴリの距離とする．

一方，チャンクタグ B，I，O を含む固有表現タグ，非固有表現タグについては，これらの間に階層的な関係が定義されていない．そのため，これらのタグに対しても近さの値を与える関数を考える必要がある．本研究では，これを解決する手段として，前述のカテゴリ間の距離を拡張する方法を用いる．

まず，B，I が付加されたカテゴリを付加される前（PERSON 等）と同一視する写像 $\varphi : \mathcal{L}_e \to \mathcal{E}$ を次のように定義する．

$$\varphi(l_{kB}) = e_k, \quad \varphi(l_{kI}) = e_k \tag{6}$$

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 $l, l' \in \mathcal{L}$ に対するコスト関数 $c_{\text{hie}} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}_+$ を改めて式 (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，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 を「d のとりうる最大値 $+1$」とした．

カテゴリ間の距離を与える関数を拡張し，ラベル間のコスト関数 c_{hie} を定義した．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．コスト関数を導入した評価関数を学習に用いることで，より誤りの程度の小さい固有表現抽出の実現を目指す．

4.3 本手法で用いる素性

本手法では，単語列 $\boldsymbol{x}$ およびタグ列 $\boldsymbol{y}$ 中の同じ位置 t における単語とタグの組 (x_t, y_t) と，タグ列中の連続する位置 $t-1, t$ にあるタグの組 (y_{t-1}, y_t) を想定した素性を用いる．

5. 評価実験

5.1 実験データ

実験には，GENIA コーパス v3.02 を使用した[*2]．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等をタグ付けしたコーパスである．タンパク質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[*3]．

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” 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 が付与されている場合には，単語列 “IL-2 gene expression” は固有表現 other_name であるとした．

[*2] `http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/corpus/GENIAcorpus3.02.tgz` から入手可能．
[*3] 文献 [10] 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．

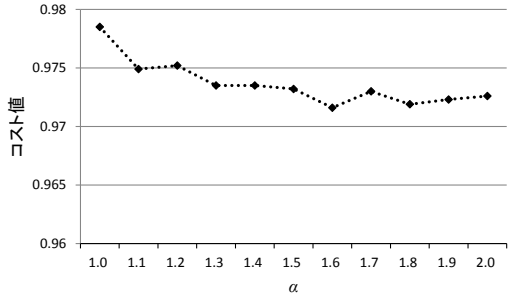

図 2 階層コスト関数（$1 \leq \alpha \leq 2$）におけるコスト値
Fig. 2 Cost function value when using the c_{hie} cost function ($1 \leq \alpha \leq 2$).

5.2 評価方法

GENIA コーパスで 10 分割交差検定を行い，10 回のコスト値の平均を評価に用いた．コスト値とは，テスト事例に対するコスト関数 C（ただし，式 (4) 中の c としては式 (7) の c_{hie} を用いる）の値を全事例について足し合わせ，出現単語数で割って正規化した値である．

コスト値は，テストデータ中の全単語に対するコスト関数 c_{hie} 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，コスト値は階層性を考慮した評価尺度となっている．コスト値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．

5.3 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．α の値を変化させながら，各値に対してそれぞれ 10 分割交差検定を行い，最もコスト値が良くなる値を最適な値とした．学習は，評価値が収束するまで 50 回行った．結果を**図 2** に示す．$\alpha = 1.6$ において，コスト値 0.9716 で最良となっており，この値を最適な α の値として以後の実験で用いる．

5.4 コスト関数の違いによる精度の比較

4.2 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[*4]．

[*4] 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．

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

	コスト値
ベースライン	1.0237
階層コスト関数（$\alpha = 1.6$）	**0.9716**
Hamming コスト関数（$\alpha = 14.0$）	0.9991

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

本実験では，コスト関数を用いない通常の評価関数を学習に用いるベースライン，階層コスト関数 c_{hie} を評価関数に導入した場合，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，コスト値を算出した．Hamming コスト関数における最適な α の値は，5.3 節の階層コスト関数の場合と同様に決定した．実験結果を**表 1** に示す．

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 つの組み合わせについて，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わかる．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．

5.5 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 [12] と比較を行った．

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カテゴリに分類している．10 分割交差検定を行い，F 値 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，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 値をそれぞれ算出した．結果を**表 2** に示す．

本提案手法が 3 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 2 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タスクを定式化しており，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

表 2　関連研究との比較
Table 2　Comparison with exsisting methods.

	F 値
構造化パーセプトロン	68.7
構造化パーセプトロン+階層コスト関数（$\alpha = 1.6$）	**69.8**
Lee et al., 2004	66.7

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．

また，ベースラインに対して，階層コスト関数を用いることで F 値が向上している．こちらもコスト値と同様，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わかる．

6. まとめと今後の課題

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．

本手法では，同じ位置に存在するトークンとラベルの組など，タスクやデータセットに依存しない一般的な素性のみを用いて推定を行った．そのため，固有表現抽出に特化した素性を用いることで，さらに抽出精度が向上しうると考えられる．一方，ドメインに特化した素性を用いずに，コスト関数を用いた方法が有効であることを確認した．このことから，本手法は固有表現抽出だけでなく，一般的な系列ラベリングに適用した場合にも有効である可能性がある．今後の課題として，ドメインに特化した素性を導入して，より高精度な固有表現抽出の実現を目指すとともに，ラベルが階層構造を持つ系列ラベリングの他のデータセットも用いて，本手法の有効性をさらに検証していきたい．

参考文献

[1] Chieu, H. and Ng, H.: Named entity recognition with a maximum entropy approach, *Proceedings of CoNLL-2003*, pp. 160–163 (2003).
[2] Collins, M.: Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms, *Proceedings of the 2002 conference on EMNLP*, pp. 1–8 (2002).
[3] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R.: The automatic content extraction (ACE) program–tasks, data, and evaluation, *Proceedings of the 4th conference on LREC*, pp. 837–840 (2004).
[4] Finkel, J., Grenager, T. and Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling, *Proceedings of the 43rd Annual Meeting on ACL*, pp. 363–370 (2005).
[5] Fleischman, M.: Automated subcategorization of named entities, *Proceedings of the 39th annual meeting of the ACL*, pp. 25–30 (2001).
[6] Fleischman, M. and Hovy, E.: Fine grained classification of named entities, *Proceedings of the 19th international conference on computational linguistics*, pp. 1–7 (2002).
[7] Florian, R., Ittycheriah, A., Jing, H. and Zhang, T.: Named entity recognition through classifier combination, *Proceedings of CoNLL-2003*, pp. 168–171 (2003).
[8] Freund, Y. and Schapire, R.: Large margin classification using the perceptron algorithm, *Machine learning*, Vol. 37, No. 3, pp. 277–296 (1999).
[9] Grishman, R. and Sundheim, B.: Message understanding conference-6: A brief history, *Proceedings of the 16th International Conference on COLING*, pp. 466–471 (1996).
[10] Kim, J., Ohta, T., Tateisi, Y. and Tsujii, J.: GENIA corpus–a semantically annotated corpus for bio-textmining, *Bioinformatics*, Vol. 19, Suppl.1, pp. i180–i182 (2003).
[11] Kim, J., Ohta, T., Tsuruoka, Y., Tateisi, Y. and Collier, N.: Introduction to the bio-entity recognition task at JNLPBA, *Proceedings of the international joint workshop on NLPBA*, pp. 70–75 (2004).
[12] 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).
[13] Ohta, T., Tateisi, Y. and Kim, J.: The Genia corpus: An annotated research abstract corpus in molecular biology domain, *Proceedings of the 2nd international conference on human language technology research*, pp. 82–86 (2002).
[14] 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).
[15] Sekine, S. and Isahara, H.: IREX: IR and IE Evaluation project in Japanese, *Proceedings of the 2nd international conference on LREC*, pp. 1475–1480 (2000).
[16] Sekine, S. and Nobata, C.: Definition, dictionaries and tagger for extended named entity hierarchy, *Proceedings of the 4th conference on LREC*, pp. 1977–1980 (2004).
[17] Sekine, S., Sudo, K. and Nobata, C.: Extended named entity hierarchy, *Proceedings of the 3rd conference on LREC*, pp. 1818–1824 (2002).
[18] Tjong Kim Sang, E. F. and De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, *Proceedings of CoNLL-2003*, pp. 142–147 (2003).
[19] Tsochantaridis, I., Hofmann, T., Joachims, T. and Altun, Y.: Support vector machine learning for interdependent and structured output spaces, *Proceedings of the 21st ICML* (2004).