Support Vector Machineを用いた日本語固有表現抽出

山田寛康[†] 工藤拓[†] 松本裕治[†]

本稿では，機械学習アルゴリズム Support Vector Machine（SVM）を用いて日本語固有表現抽出を学習する手法を提案し，抽出実験によりその有効性を検証する．固有表現抽出規則の学習には，単語自身，品詞，文字種などを素性として使用するため，その素性空間は非常に高次元となる．SVMは汎化誤差が素性空間の次元数に依存しないため，固有表現抽出規則の学習においても過学習を起こすことなく汎化性能の高い学習が実現できる．また多項式 Kernel 関数を適用することで複数の素性の組合せを考慮した学習が計算量を変えることなく実現できる．CRL 固有表現データを用いて IREX 固有表現抽出タスクに対して実験を行った結果，語彙，品詞，文字種，およびそれら任意の 2 つの組合せを考慮した場合，交差検定により F 値で約 83 という高精度の結果が得られた．

Japanese Named Entity Extraction Using Support Vector Machine

Hiroyasu Yamada,[†] Taku Kudo[†] and Yuji Matsumoto[†]

In this paper, we propose a method for Japanese named entity (NE) extraction using Support Vector Machines (SVM). The generalization performance of SVM does not depend on the size of dimensions of the feature space, even in a high dimensional feature space, such as named entity extraction task using lexical entries, part-of-speech tags and character types of words as the primitive features. Furthermore, SVM can induce an optimal classifier which considers the combination of features by virtue of polynomial kernel functions. We apply the method to IREX NE task using CRL Named Entities data. The cross validation result of the F-value being 83 shows the effectiveness of the method.

1. はじめに

人名・組織名といった語句を同定する固有表現抽出タスク（Named Entity Extraction）は，情報検索や情報抽出の基礎技術としてのみならず，自然言語処理における形態素解析や構文解析などに大きな影響を及ぼすため，重要な問題である．近年，英語の固有表現抽出は Message Understanding Conference（MUC-7）[3] などでさかんに行われており，また日本では，Information Retrieval and Extraction Exercise（IREX）[2] において日本語を対象とした固有表現抽出タスクが行われている．

固有表現を抽出する手法として，人手により抽出規則を作成し，それらの規則を対象テキストに適用することで固有表現を抽出する手法がある．この方法は高精度の抽出規則が期待できる反面，新しく出現した固有表現に対してこれまでの規則を変更したり新たな規則を追加したりする必要があるため，多大なコストを要する．

これに対して，固有表現がタグ付けされたテキストから，最大エントロピー法[13]，決定木学習[4]，あるいは決定リスト[10),15)] などの機械学習を用いて抽出規則を自動的に学習する手法がある．これらの手法は，固有表現がタグ付けされたテキストさえあれば，新たなテキストに対しても抽出規則の生成に関して人手を必要としない利点がある．

精度の良い固有表現抽出規則を学習するためには，学習に用いる素性として，単語自身，品詞，あるいは文字種など様々な素性を用いる．したがって，その素性数は数万以上となり高次元素性空間での学習が必要となる．一般に素性空間の次元数が増加するにつれて過学習を引き起こしやすくなるため，高次元素性空間においてもできるだけ過学習を起こさない学習アルゴリズムが必要となる．

Support Vector Machine[8] は，その汎化誤差が素性空間の次元数に依存しないことが理論的に証明されており，実験的にも Chunking[5]，文書分類[6),9),14)] などの高次元素性空間を用いる学習で高い精度が報告されている．また多項式 Kernel 関数を適用することで，複数の素性の組合せを考慮した学習が計算量を大きく変えることなく実現可能である．

† 奈良先端科学技術大学院大学情報科学研究科
The Graduate School of Information Science of Nara Institute of Science and Technology

表 1 IREX で使用する固有表現の種類と例
Table 1 The definition of Japanese named entity in IREX and it's examples.

固有表現の種類		例
ARTIFACT	固有物名	ノーベル文学賞
DATE	日付表現	五月五日
LOCATION	地名	日本，韓国
MONEY	金額表現	２０００万ドル
ORGANIZATION	組織名	社会党
PERCENT	割合表現	二〇%，三割
PERSON	人名	村山富市
TIME	時間表現	午前五時

本稿では，IREX 日本語固有表現抽出タスク[2] に対して，Support Vector Machine を用いて固有表現抽出規則を学習し，抽出実験によりその有効性を検証する．

以下，次章では IREX 日本語固有表現抽出について述べる．3 章で学習アルゴリズム Support Vector Machine について説明し，4 章で固有表現抽出に Support Vector Machine を適用する方法について説明する．5 章で抽出実験と考察について報告し，最後に 6 章でまとめと今後の課題について述べる．

2. IREX 日本語固有表現抽出

IREX 日本語固有表現抽出タスクでは**表 1** に示す 8 種類の固有表現を定義し，それぞれの固有表現は入れ子にはならないとしている．固有表現抽出は，入力文中の単語列が固有表現か否かを識別する Chunk 同定問題と見なすことができ，Chunk 同定問題では 1 つ以上の要素列からなる Chunk を IOB1，IOB2，IOE1，および IOE2 という 4 種類のタグを使用して表記する手法が提案されている[1]．これとは別に，内元らは複数の単語列からなる日本語固有表現のために Start/End（SE）というタグを使用した表記法を提案している[13]．本研究でも固有表現抽出のためにこれらの表記を使用する．

図 1 は “エリツィン大統領は四日，日米両国 ···” という文中で，エリツィン：人名 (PERSON)，四日：日付 (DATE)，日：地名 (LOCATION)，米：地名 (LOCATION) という 4 つの固有表現に対して，IOB1，IOB2，IOE1，IOE2，および SE のそれぞれの記法による違いを表している．IOB1 は固有表現である単語に対して I というタグを付与し，同種類で別の固有表現が連続した場合は，後続する固有表現の開始単語に B というタグを付与する．固有表現以外の単語は O というタグを付与する．IOB2 は，開始位置に付与するタグが IOB1 とは異なり，固有表現の開始単語に必ず B というタグを付与する．IOB1 と IOB2 が開始位置に注目するのに対して，IOE1 と IOE2 はそれぞれ，終了位置に注目し，E というタグを付与する．SE は，1 つの単語からなる固有表現に S というタグを付与する．複数の単語からなる固有表現には，その開始単語に B，終了単語に E，固有表現内の単語に I，そして固有表現以外の単語に O というタグを付与する．

以降本稿では，混乱を避けるために，固有表現の開始終了位置を表す B，I，O，E，S の表記を **Chunk タグ**と呼び，IREX で定義した 8 つの固有表現を**固有表現の種類**と呼ぶ．そして Chunk タグと固有表現の種類が 1 つになった，B-DATE のような表記を**固有表現タグ**と呼ぶ．

固有表現タグを使用することで，固有表現抽出規則の学習は，入力文中の各単語を固有表現タグに分ける分類規則の学習として扱うことが可能となる．

3. Support Vector Machine

図 2 は Support Vector Machine（SVM）の概要図を示す．SVM は，n 次元素性ベクトル $\mathbf{x}_t$ と正・負のラベル y_t のペア $(\mathbf{x}_t, y_t)$ で表現される l 個の訓練事例 $(0 < t \leq l)$ に対して，正・負例を正しく分離する超平面 $\mathbf{w} \cdot \mathbf{x} + b, (\mathbf{w}, \mathbf{x} \in \mathbf{R}^n)$ を求める二値線形分類器である[8]．図 2 において，破線は求める分離超平面に平行で等距離にある超平面で，この間の距離をマージンと呼ぶ．SVM は正・負例を正しく分離する数多くの超平面の中から，マージンが最大となる分離超平面を求めるアルゴリズムである．マージンの最大化は $\|\mathbf{w}\|$ の最小化であり，これは式 (1) を式 (2) の条件で最大化する双対問題と等価であることが知られている．

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \tag{2}$$

$$K(\mathbf{a}, \mathbf{b}) = \mathbf{\Phi}(\mathbf{a}) \cdot \mathbf{\Phi}(\mathbf{b}) \tag{3}$$

ここで式 (1) の $K(\mathbf{x}_i, \mathbf{x}_j)$ を Kernel 関数と呼び，式 (3) で示す 2 つのベクトル $\mathbf{a}, \mathbf{b} \in \mathbf{R}^n$ を関数 $\mathbf{\Phi}(\mathbf{x})$ で写像した空間での内積を表す．最終的に未知の事例 $\mathbf{x}$ に対する正・負例の分類は，超平面からの位置（式 (4) の関数値が正ならば正例，負であれば負例）により決定される．

$$\begin{aligned} f(\mathbf{x}) &= sgn\left(\mathbf{w} \cdot \mathbf{x} + b\right) \\ &= sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \end{aligned} \tag{4}$$

	エリツィン	大統領	は	四	日	,	日	米	両国
IOB1	I-PERSON	O	O	I-DATE	I-DATE	O	I-LOCATION	B-LOCATION	O
IOB2	B-PERSON	O	O	B-DATE	I-DATE	O	B-LOCATION	B-LOCATION	O
IOE1	I-PERSON	O	O	I-DATE	I-DATE	O	I-LOCATION	E-LOCATION	O
IOE2	E-PERSON	O	O	I-DATE	E-DATE	O	E-LOCATION	E-LOCATION	O
SE	S-PERSON	O	O	B-DATE	E-DATE	O	S-LOCATION	S-LOCATION	O

図 1 固有表現タグ
Fig. 1 Named entity tag.

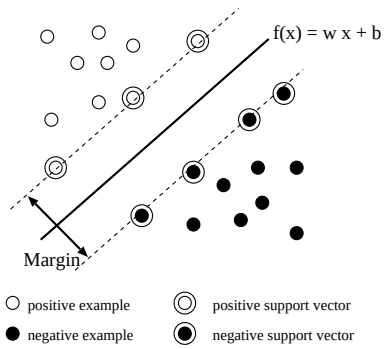

図 2 Support Vector Machine による分離超平面
Fig. 2 The decision surface of a linear Support Vector Machine.

3.1 SVM の汎化誤差

Vapnik らは，l 個の訓練事例で学習した仮説 f の汎化誤差 R が，$1-\eta$ の確率で式 (5) を満たすことを証明している[8]．ここで R_{emp} は訓練事例に対するエラー率であり，h は VC 次元と呼ばれる学習モデルの複雑さを表す指標である．l と R_{emp} が一定であると仮定すると，h を最小にすることで R の上限値を最小化できる．SVM の VC 次元 h は，事例を覆う最小超球の直径 D，マージン ρ と素性空間の次元数 n により式 (6) に示す上限値の存在が証明されている[8]．ここで n が十分に大きい場合，ρ の最大化が VC 次元を最小化し，結果的に高次元素性空間でも式 (5) の汎化誤差の上限を最小にすることが可能である．

$$R \leq R_{emp} + \sqrt{\frac{h(\ln(2l/h)+1) - \ln(\eta/4)}{l}} \quad (5)$$

$$h \leq \min(D^2/\rho^2, n) + 1 \quad (6)$$

3.2 Kernel 関数の適用よる素性の組合せを考慮した学習

SVM は式 (3) で示される Kernel 関数を適切に選択することで非線形分離問題に対応可能である．特に Kernel 関数として d 次の多項式関数を使用することで d 個までの素性の組合せを考慮した学習が，計算量を大きく変化させることなく可能となる．

例として 2 次元の素性ベクトル $\mathbf{a} = (a_1, a_2)$，$\mathbf{b} = (b_1, b_2)$ に対し，式 (7) で示す二次の多項式 Kernel 関数を適用する場合を考える．2 次元の素性ベクトル $\mathbf{x} = (x_1, x_2)$ を，式 (9) に示す 6 次元空間へ写像する関数 $\mathbf{\Phi}(\mathbf{x})$ を考える．$\mathbf{\Phi}(\mathbf{x})$ で写像した空間の成分は，元の 2 次元空間における各成分の 2 次までの項で表され，これは素性の 2 つまでの組合せを考慮した空間といえる．

式 (7) は，最終的に式 (8) のように表すことができ，これは ベクトル $\mathbf{a}$，$\mathbf{b}$ それぞれを，$\mathbf{\Phi}(\mathbf{x})$ によって写像した 6 次元空間での内積の値を表している．すなわち Kernel 関数 $(\mathbf{a}\cdot\mathbf{b}+1)^2$ の値を求めることで，6 次元空間での内積の値が，ベクトル $\mathbf{a}$，$\mathbf{b}$ を明示的に $\mathbf{\Phi}(\mathbf{x})$ で写像することなく求めることができる．

$$\begin{aligned} K(\mathbf{a},\mathbf{b}) &= (\mathbf{a}\cdot\mathbf{b}+1)^2 \quad (7)\\ &= (a_1b_1 + a_2b_2 + 1)^2 \\ &= a_1^2b_1^2 + 2a_1a_2b_1b_2 + 2a_1b_1 \\ &\quad +2a_2b_2 + a_2^2b_2^2 + 1 \\ &= \mathbf{\Phi}(\mathbf{a})\cdot\mathbf{\Phi}(\mathbf{b}) \quad (8) \end{aligned}$$

$$\begin{aligned} \mathbf{\Phi}(\mathbf{x}) &= (x_1^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, x_2^2, 1), \quad (9)\\ \mathbf{x} &= (x_1, x_2) \end{aligned}$$

また n 次元ベクトル $\mathbf{x}_i$，$\mathbf{x}_j$ に対し，d 次の多項式関数を Kernel 関数として使用した場合にも同様なことがいえ，SVM に Kernel 関数を適用することにより，$\mathbf{\Phi}(\mathbf{x})$ で写像した空間において，線形分離問題を解くことと等価となる．このとき，内積 $\mathbf{\Phi}(\mathbf{x}_i)\cdot\mathbf{\Phi}(\mathbf{x}_j)$ の値は，Kernel 関数を使用することで，明示的に $\mathbf{\Phi}(\mathbf{x})$ で写像することなしに計算が可能なため，計算量は大きく増加しない．

多項式関数以外の代表的な Kernel 関数としては，Sigmoid Kernel，RBF Kernel などがある．しかしこれらの Kernel 関数は，固有表現抽出規則の学習において，多項式関数のような組合せを考慮した学習といった直観的解釈が困難であることから，本稿では実験の対象外とした．

3.3 二値分類から多値分類への拡張

SVM は正例・負例を分類する二値分類器であり，固有表現抽出規則を学習するためには 3 つ以上のクラスに分類する多値分類に拡張する必要がある．代表的な手法として one class vs. all others 法と pairwise 法

がある．one class vs. all others 法は，あるクラスかそれ以外かという二値分類器を分類するクラス数分構築する方法で，pairwise 法は k 個のクラスから任意の 2 つのクラスに関する二値分類器を ${}_kC_2$ 個構築する方法である．本稿では手書き文字認識において高精度な結果が報告されている pairwise 法を用いた[7]．

pairwise 法について説明する．今，クラス s とクラス t を分類する二値分類器 $f_{st}(x)$ を考える．$f_{st}(x)$ は，$f_{st}(x) \geq 0$ のとき，事例 $\mathbf{x}$ をクラス s と判定し，反対に $f_{st}(x) < 0$ のときクラス t と判定する．クラス c の投票数を，${}_kC_2$ 個ある二値分類器のうちクラス c と判定した分類器の個数とする．pairwise 法での最終的な分類クラスは，投票数が最も多いクラスに決定される．

4. SVM の日本語固有表現抽出への適用

SVM を日本語固有表現抽出タスクに適用する方法について説明する．

固有表現抽出規則の学習は，入力された文の各単語に対し，固有表現タグに分類する規則を学習することである．また固有表現抽出は，未知の文に対して，各単語の固有表現タグを推定することである．そのため，まず文を形態素解析し，単語列に分割する．固有表現タグの学習，および固有表現タグの推定を行う単位は単語単位であり，1 つの事例は 1 単語に対応する．このとき，文頭から順に固有表現タグを推定する方法を**右向き解析**と呼び，逆に文末から順に推定する方法を**左向き解析**と呼ぶ．右向き解析と左向き解析とでは，学習および抽出時に使用する素性が異なるため，これら 2 つの方法を区別し実験を行う．

固有表現抽出規則の学習

右向き解析を行う場合の学習について説明する．文頭から i 番目の単語に関する素性は $i-2$ から $i+2$ 番目までの各単語の，単語自身，品詞，および文字種を使用する．また複数の単語からなる固有表現を考慮するために，$i-2$ と $i-1$ 番目の固有表現タグ（学習時は既知）も素性として使用する．これらの素性を要素とするベクトル $\mathbf{x}$ と，i 番目の固有表現タグを，分類すべきクラス y とすれば，$(\mathbf{x}, y)$ という組が 1 つの事例となる．

左向き解析を行う場合の学習は，使用する素性の種類については右向き解析と同様である．しかし文末を始点とするため i に対する位置が右向き解析とは逆になる．$i-n(n>0)$ は i に対して文末側の n 個隣の単語を表し，$i+n$ は i に対して文頭側の n 個隣の単語を表す．

位置	$i-2$	$i-1$	i	$i+1$	$i+2$
入力文	大統領	は	五	日	午前
品詞	名詞	助詞	名詞	名詞	名詞
文字種	漢字	平仮名	漢字	漢字	漢字
固有表現タグ	O	O	B-DATE	I-DATE	B-TIME

図 3 使用する素性
Fig. 3 An example of features used in the experiment.

固有表現抽出

i 番目の固有表現タグの推定には，学習時と同様 $i-2$ から $i+2$ 番目までの各単語の，単語自身，品詞，および文字種を素性として使用する．未知の文に対する固有表現抽出では，解析の最初では，固有表現タグは未知である．このため $i-2$ と $i-1$ 番目の固有表現タグは，各位置で推定した結果をそのまま使用した．

文全体で最適な固有表現タグの推定を行う方法として，ビタビアルゴリズムなどの動的計画法がある．しかしこの方法では，各固有表現タグの推定において適切な尤度を計算する必要がある．SVM は，ある単語が，ある固有表現タグとなる尤度を出力しない．pairwise 法を用いた場合，投票数を尤度として解釈できるが，適切な尤度であるという理論的な根拠はない．よって本稿では，各固有表現タグの推定結果をそのまま使用し，決定的に解析を行う方法を採用した．

図 3 は，入力文 “大統領は五日午前 ···” に対して，右向き解析の場合に使用する素性の例を示す．学習時において，単語 “五” に関する事例は，分類するクラスは B-DATE で，素性は，枠で囲まれた要素すべてを使用する．同様の文をテストデータとし固有表現抽出する場合，“五” の素性は学習時と同様に枠内の要素すべてを使用する．$i-2$ と $i-1$ 番目の固有表現タグは，それぞれの位置で SVM によって推定した結果をそのまま使用する．推定した “五” の固有表現タグは，以後の “日” と “午前” の固有表現タグ推定に，素性として使用する．

5. 実　　験

5.1 デ　ー　タ

実験には CRL（郵政省通信総合研究所）固有表現データを使用した☆．CRL 固有表現データは，毎日新聞 95 年度版 1,174 記事，約 11,000 文に対して IREX で定義された固有表現がタグ付けされている．このデータ中の固有表現の総数は 19,262 個であった．形

☆ 現在，IREX 日本語固有表現抽出タスク本試験データは参加者以外使用できない．そのため本稿では非参加者が使用可能でデータ量の多い CRL 固有表現データで実験を行った．

表 2 固有表現の種類その頻度
Table 2 The frequency distribution of named entity in the CRL data.

固有表現の種類	出現頻度	(%)
ARTIFACT	747	(3.88)
DATE	3,567	(18.52)
LOCATION	5,463	(28.36)
MONEY	390	(2.02)
OPTIONAL	585	(3.04)
ORGANIZATION	3,676	(19.08)
PERCENT	492	(2.55)
PERSON	3,840	(19.94)
TIME	502	(2.61)
合計	19,262	

態素解析器は茶筌[12] を使用し，評価は CRL 固有表現データを 5 等分に分割し，訓練 4，テスト 1 の比率で交差検定を行い，それらの総合の F 値 ($\beta = 1$) を使用した．**表 2** に CRL 固有表現データ中に出現した各固有表現の頻度とその割合を示す．OPTIONAL は人手によって正解固有表現タグを付与するときに，一意に決めることが困難であった場合に付与されるタグである．IREX では OPTIONAL について，その開始終了位置を誤って推定をした場合のみ不正解としている．本稿でもこの定義に従った．また，実験では Kernel 関数として，d 次の多項式関数 $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ を使用した．

5.2 素性の違いによる精度の比較

はじめに，Chunk タグを IOB2，解析方向を右向き，Kernel 関数は二次の多項式関数に固定し，抽出精度を，使用する素性が以下に示す 4 種類の場合について調査した．

(1) 単語自身，品詞大分類
(2) 単語自身，品詞細分類
(3) 単語自身，品詞大分類，文字種
(4) 単語自身，品詞細分類，文字種

ここで品詞大分類とは，名詞，動詞，助詞などの分類であり，品詞細分類とは，名詞–普通名詞，動詞–自立–サ変などのより細かい分類である．文字種は “カタカナ”，“平仮名”，“漢字”，“記号”，“数字”，および “アルファベット” の 6 種類とし，単語に含まれる文字種すべてを素性として使用した．

結果を**表 3** に示す．素性空間の次元数が最も高い (4) について最良の結果が得られた．これは SVM の素性空間の次元数に依存しない高い汎化能力の実証といえる．また表 3 より，地名 (LOCATION)，組織名 (ORGANIZATION)，および人名 (PERSON) の 3 つの固有表現は，品詞細分類情報を使用することにより，大幅な精度の向上がみられた．茶筌の名詞細分類情報には，名詞–固有名詞–人名など，これら 3 種類の固有表現そのものを表す情報が付与されているためである．この 3 種類の固有表現を精度良く抽出するためには，ある程度の固有名詞に関する情報が必要である．日付表現 (DATE)，および割合表現 (PERCENT) の 2 つの固有表現に関しては，品詞細分類情報を使用した場合と，使用しない場合では大きな差はなかった．残りの固有物名 (ARTIFACT)，金額表現 (MONEY)，および時間表現 (TIME) は，品詞細分類情報を使用した場合，使用しない場合に比べ若干精度が低下した．今後，各固有表現別に最適な素性を選択することで，より高精度の抽出が可能になると考えられる．

表 3 素性の違いによる抽出精度
Table 3 The extraction accuracy for each feature set.

	$F_{\beta=1}$ 値			
素性の種類	(1)	(2)	(3)	(4)
ARTIFACT	46.5	45.8	**47.0**	45.9
DATE	89.4	91.1	90.7	**91.4**
LOCATION	76.3	81.6	78.1	**82.1**
MONEY	90.5	91.3	**91.5**	91.2
ORGANIZATION	64.2	75.8	67.8	**76.1**
PERCENT	91.2	91.3	91.3	**91.9**
PERSON	69.6	83.7	72.9	**85.0**
TIME	89.0	88.9	**89.3**	88.9
総合	74.8	81.5	76.8	**82.0**

5.3 Chunk タグと解析方向の違いによる精度の比較

次に素性を，単語自身，品詞細分類，および文字種に固定し，Chunk タグと解析方向の違いによる抽出精度を調査した．**表 4** に結果を示す．最良の精度は IOB2 の左向き解析の場合で 83.2 であった．また IOB1 を除くすべての Chunk タグで，右向きよりも左向き解析のほうが良い結果が得られた．これは，複数の単語からなる固有表現は，その後方の単語によって種類が決定される場合が多いことが原因だと考えられる．たとえば「野村証券」という固有表現が組織名であることは,「野村」という語ではなく「証券」という語に強く起因する．本稿では各解析方向で決定的に固有表現タグを推定するため，左向き解析では「証券」の固有表現タグを先に推定し，その推定結果を「野村」の固有表現タグ推定に素性として使用する．このため後方の単語の推定結果が固有表現全体の推定に強く影響し，右向き解析よりも良い結果が得られたと考えられる．

5.4 Kernel 関数の違いによる精度の比較

次に，適用する多項式 Kernel 関数の次数 d を 1 から 4 に変化させ，素性の組合せを考慮した学習が，固有表現抽出にどれだけ重要であるかを調査した．素性として，単語自身，品詞細分類，および文字種を使用

表 4 Chunk タグの違いと解析方向の違いによる抽出精度
Table 4 A combination of type of chunk tag and parsing direction and the extraction accuracy.

	$F_{\beta=1}$ 値									
タグの種類	IOB1		IOB2		IOE1		IOE2		SE	
解析方向	右	左	右	左	右	左	右	左	右	左
ARTIFACT	47.1	44.5	46.8	47.1	47.2	**48.3**	44.8	46.3	44.1	47.1
DATE	92.0	91.9	91.9	92.2	91.9	**92.6**	91.3	92.4	91.0	91.8
LOCATION	82.6	82.3	82.3	82.5	82.5	**82.8**	82.1	81.9	81.7	81.8
MONEY	90.7	93.9	91.0	**94.3**	90.7	93.9	90.6	94.1	90.9	94.0
ORGANI.	76.1	78.0	75.7	**79.0**	76.3	76.7	75.2	75.7	74.7	77.9
PERCENT	89.2	**94.8**	91.1	94.2	89.1	92.5	88.6	91.8	91.4	93.7
PERSON	85.0	85.5	85.0	**86.3**	85.1	85.5	84.9	85.2	84.8	85.7
TIME	87.1	89.0	88.7	**89.2**	87.1	86.5	84.3	86.3	88.7	88.7
総合	82.4	82.1	82.3	**83.2**	82.4	82.9	81.2	81.7	81.2	82.4

表 5 多項式 Kernel 関数の次数による抽出精度
Table 5 The number of degree of polynomial kernel function and the extraction accuracy.

	$F_{\beta=1}$ 値			
d	1	2	3	4
ARTIFACT	28.7	**47.1**	46.3	44.3
DATE	89.1	**92.2**	91.2	90.2
LOCATION	78.4	**82.5**	81.9	81.5
MONEY	94.1	**94.3**	93.7	93.5
ORGANIZATION	71.5	**79.0**	77.5	76.7
PERCENT	92.8	**94.2**	93.1	92.8
PERSON	82.0	**86.3**	85.6	85.2
TIME	86.5	**89.2**	87.4	86.9
総合	78.5	**83.2**	82.3	81.7

表 6 前 2 単語の固有表現タグを素性として使用する効果
Table 6 Effect of NE extraction using previous NE tag as features.

	$F_{\beta=1}$ 値	
固有表現タグ	使用する	使用しない
ARTIFACT	**46.5**	23.6
DATE	**91.9**	86.9
LOCATION	**84.0**	81.4
MONEY	**87.8**	77.2
ORGANIZATION	**79.4**	68.7
PERCENT	91.5	**92.1**
PERSON	**85.4**	82.3
TIME	**85.2**	76.6
総合	**82.8**	77.2

し，Chunk タグは IOB2，解析方向は左向きで実験を行った．

結果を**表 5** に示す．$d = 1$，すなわち素性の組合せを考慮しない場合は，$d > 1$ で素性間の組合せを考慮した場合と比べて大きく劣ることが分かる．日本語固有表現抽出では，単語自身，品詞，文字種などを独立に学習するのではなく，それらの組合せを考慮することが重要であるといえる．また任意の 2 つの素性の組合せを考慮した 2 次の多項式の場合が最良の精度であった．3 次以上の多項式では，訓練事例数に対して必要以上に素性空間の次元数が増加したため，汎化性能が低下し，抽出精度が低下したと考えられる．

5.5 ビームサーチ法による固有表現抽出との比較

本稿では，各単語の固有表現タグ推定を決定的に行うことで固有表現を抽出した．しかし決定的な解析では，誤った推定結果が後方の解析に悪影響を及ぼし，結果的に精度を低下させる危険性がある．そこで，まず学習および分類に使用する素性として，前 2 単語の固有表現タグを使用しない場合，抽出精度にどの程度影響を与えるかを調査した．実験は交差検定の 1 つのセットに対して行い，使用する素性は，前 2 単語の固有表現タグを使用しないことを除いては 5.4 節と同じものとした．また多項式 Kernel 関数の次数 d は 2 とし，Chunk タグを IOB2，解析方向を左向き解析という設定で行った．結果を**表 6** に示す．全体の抽出精度は F 値で 77.2 であった．同一の交差検定のセットに対し，前 2 単語の固有表現タグを素性として使用した場合に比べ，PERCENT 以外のすべての固有表現において，精度が大きく低下した．決定的な解析方法を用いても，前 2 単語の固有表現タグを素性として使用することで，精度向上に大きく貢献することが分かった．

次に，文全体でより適切な固有表現タグの推定を行うために，ビームサーチ法を用いて解析し，決定的に固有表現タグを推定する本手法との比較を行った．実験は交差検定のすべてのセットに対し行い，使用する素性は 5.4 節と同じものとした．また多項式 Kernel 関数の次数 d は 2 で，IOB2，左向き解析という設定で行った．ビームサーチ法で使用する尤度は，pairwise 法における各固有表現タグの投票数とし，ビーム幅を 3，5，10 と変化させ抽出精度を調査した．

結果を**表 7** に示す．表 7 において，解析時間は，決

表 7 ビーム幅の変化による抽出精度
Table 7 The width of beam search and the extraction accuracy.

	$F_{\beta=1}$ 値			
ビーム幅	1	3	5	10
ARTIFACT	47.1	**47.3**	46.3	46.6
DATE	92.2	**92.3**	**92.3**	**92.3**
LOCATION	82.5	82.6	82.6	**82.7**
MONEY	**94.3**	**94.3**	**94.3**	**94.3**
ORGANIZATION	**79.0**	78.8	78.8	78.7
PERCENT	94.2	94.2	94.3	**94.5**
PERSON	**86.3**	86.0	86.0	85.9
TIME	**89.2**	89.1	88.8	89.0
総合	**83.2**	**83.2**	83.1	**83.2**
解析時間	1	3.42	5.52	10.73

定的に固有表現抽出を行う方法での 1 文あたりの平均解析時間を 1 としたときの比を示す☆．表 7 より，ビーム幅を広げるにつれ解析時間は増加するが，全体の抽出精度は向上しなかった．この原因の 1 つに，今回尤度として定義した pairwise 法の投票数が，適切な尤度でないことが考えられる．今後，SVM を用いて文全体で最適な固有表現タグの推定を行うためには，理論的根拠に基づく尤度の定義は重要な課題である．

5.6 誤りの原因

本節では固有表現抽出誤りの主な原因について述べる．

5.6.1 形態素解析の単語分割による誤り

日本語固有表現抽出において，単語を単位に固有表現タグを推定する場合，前処理として形態素解析により文を単語に分割する必要がある．この場合，形態素解析によって分割された単語単位では，正しく固有表現タグを付与できない場合がある．例を**図 4** に示す．形態素解析によって名詞 1 語と判断された “訪中” は “中” の部分だけが地名を表す固有表現である．したがって “中” だけに B-LOCATION などの固有表現タグを付与することができない．この問題に対処するには，“訪中” を “訪” と “中” の 2 つに細かく分割する必要がある．

訓練データは，どの単語が固有表現であるかが既知である．したがって形態素解析が，固有表現の開始直前，および終了直後で必ず単語分割するように前処理を行うことで，誤った固有表現タグの付与は起こらない．これに対してテストデータでは，どの単語が固有表現であるかは未知であるため，形態素解析が，固有表現の開始直前および，終了直後で単語分割せず，正

☆ 実験に使用した計算機の CPU は Pentium III 933 MHz で，ビーム幅 1 のときの 1 文平均解析時間は約 0.91 秒であった．

形態素解析による単語分割

	名詞	名詞	助詞	名詞		助詞	
···	三	日	に	訪	中	し	···
	日付		-	-	地名	-	

固有表現

図 4 形態素解析の単語分割と固有表現の開始終了位置の相違
Fig. 4 An example of different segmentation between tokenizer and named entity in Japanese.

表 8 単語分割の違いによる抽出精度
Table 8 Three tokenization methods and the extraction accuracy.

	$F_{\beta=1}$ 値		
	(A)	(B)	(C)
ARTIFACT	47.1	**48.3**	48.2
DATE	92.2	92.7	**92.8**
LOCATION	82.5	82.6	**87.8**
MONEY	94.3	94.3	**94.5**
ORGANIZATION	79.0	77.3	**81.4**
PERCENT	94.2	91.4	**97.1**
PERSON	86.3	85.7	**86.4**
TIME	89.2	85.2	**90.1**
総合	83.2	83.0	**85.9**
適合率 (%)	86.4	85.1	88.1
再現率 (%)	80.3	81.0	83.9

しい固有表現タグの付与が不可能な場合が生じる．

我々は，このような単語分割に関する問題が，抽出精度にどの程度影響を与えるかをみるため，テストデータに対し，以下に示す (A)，(B)，および (C) の 3 種類の前処理を行った場合それぞれについて，抽出精度を調査した．

(A) テストデータに対し，形態素解析の単語分割をそのまま使用した場合．

(B) テストデータ中，訓練データに出現した固有表現と同一の単語に対し，形態素解析が固有表現の開始直前，および終了直後で必ず単語分割を行うように前処理を行った場合．

(C) テストデータ対し，正解データを使用し，出現するすべての固有表現に対し，形態素解析器が，固有表現の開始直前および終了直後の位置で，必ず単語分割が行うように前処理を行った場合．

実験はいずれも単語自身，品詞細分類，および文字種を素性とし，Chunk タグは IOB2，Kernel 関数は 2 次の多項式，解析方向は左向き解析という設定で行った．結果を**表 8** に示す．(C) の場合，F 値で 85.9 という非常に高い精度が得られた．また (B) では，(A) と比べ若干精度が低下した．一般に文字列が同じ単語でも，固有表現となるか否かは文脈に依存する．しか

表 9 関連研究との比較
Table 9 Comparison with related work.

	内元	颯々野	本手法
素性	単語自身，品詞，文字種		
形態素解析器	JUMAN	BREAKFAST	茶筌
文脈長（単語数）	±2	可変長	±2
Chunk タグ	Start/End	Start/End	IOB2
形態素解析の単語分割における問題	誤り駆動による書き換え規則の自動抽出	なし	なし
訓練データ	CRL 固有表現データ 予備試験トレーニングデータ，予備試験データ	CRL 固有表現データ	CRL 固有表現データ
サイズ（文）	約 12,000	約 11,000	約 8,800
学習法	最大エントロピー	最大エントロピー	SVM
精度（$F_{\beta=1}$ 値）	80.17	82.8	**83.2**

し (B) では，訓練データに出現した固有表現と文字列が一致する単語は，その語が固有表現の一部であるか否かにかかわらず，細かく分割されてしまう．その結果，再現率の向上に比べ，適合率が大きく低下し抽出精度が低下した．また「訪中 → 訪：未知語，中：名詞–接尾」のように，細かく分割した後のそれぞれの単語に対して，適切な品詞付与が困難な場合も原因の 1 つとして考えられる．

なお，この形態素解析の単語分割が，固有表現の開始直前および終了直後の位置で分割されない問題については，現在，国立国語研究所他による「話し言葉コーパス」プロジェクトによる粒度の細かい語単位の整備が進んでおり，茶筌でもこの辞書を採用する計画がある．この辞書の採用により上記単語分割の問題は，正しい品詞情報の付与を含め自動的に解消されると考えている．(C) の実験結果はその際の上限値を示すものと見なすことができる．

5.6.2 形態素解析器の品詞推定誤り

形態素解析器が出力した品詞が誤ったために正しい固有表現が推定できない場合があった．具体例は，地名を表す固有表現「ヒマラヤ」に対して形態素解析器が「ヒマラヤ：名詞–固有名詞–組織」という結果を出力した場合などがあげられる．品詞細分類を素性として使用することで，大幅に精度が向上する反面，抽出時に形態素解析器が品詞細分類情報を誤ると，それが不正解の直接的な原因となる．

5.6.3 未 知 語

訓練データ中に一度も出現しなかった単語を多く含む固有表現は正しい推定が難しく，特に固有物名 (ARTIFACT) は未知の単語を含む場合が多く，結果的に 8 種類の固有表現中で最も低い抽出精度であった．また単語が未知ではなくても，文学作品名やテレビ番組名などの表現は，その単語列自身が訓練データに存在しない場合，前後の単語や品詞，文字種から固有物名であるか否かを判定することは困難である．

5.6.4 組織名と地名

「ＮＴＴ」などの単語は単独では組織名を表す．しかし文脈上ある具体的な場所を示す意味合いで使用される場合，IREX の定義では地名の固有表現として抽出しなければならない．これらの 2 つの区別を正しく学習するためには，前後 2 単語の情報だけでは困難であり，係り先の動詞など情報を考慮する必要がある．

5.6.5 時間と日付表現

IREX の定義では，時間や日付表現は，文脈上ある具体的な時を表す場合にのみ固有表現として抽出する．たとえば「三日から四日」などの表現は，この日付が指すものが，特定の期日ではない期間を表す場合，固有表現としては抽出しない．本稿では素性として前後 2 単語の単語自身，品詞，文字種など表層的な情報だけを使用しているため，このような区別を正しく識別し抽出することは困難である．

5.7 関連研究との比較

本稿で提案した手法を，IREX の固有表現抽出タスクにおける他手法と比較した．比較の対象としたのは，内元らの最大エントロピー法と書き換え規則による手法[13] と颯々野らの可変長文脈を考慮した手法[15] である．**表 9** に比較対象の 2 つの研究と本手法との主な相違と精度を示す．

内元らの手法では，素性として単語自身，品詞，および文字種を使用している．形態素解析器は JUMAN[11] を使用し，素性として使用する文脈長は，前後 2 単語の固定長としている．Chunk タグは Start/End 法を使用し，形態素解析の単語分割が，固有表現の開始直前，および終了直後で分割されない問題には，誤り駆動による書き換え規則の自動抽出により対処している．学習アルゴリズムは最大エントロピー法を用いて

表 10 交差検定の各精度
Table 10 The accuracy for each data set in 5-fold cross-validation.

総合	1	2	3	4	5
83.2	82.8	83.3	**84.8**	81.6	83.6

各単語に対して固有表現タグ付与確率を推定し，ビタビアルゴリズムにより文全体で最適な固有表現タグを推定する．内元らの手法では低頻度素性の使用による過学習を回避するために頻度による素性選択を行っている．具体的には単語自身はコーパス中で 5 回以上出現したもの，単語自身以外は 3 回以上出現したものを素性として使用している．訓練データは CRL 固有表現データ，IREX-NE 予備試験トレーニングデータ，IREX-NE 予備試験データの約 12,000 文を使用し，さらに精度向上のために固有表現に関する辞書情報を使用している．最良の結果は IREX 本試験 GENERAL データに対して F 値で 80.17 と報告している．

颯々野らは，素性は内元らと同様，単語自身，品詞，および文字種を使用し，形態素解析器は BREAKFAST [16] を用いている．素性として使用する文脈長は，固有表現の長さに応じて可変長に拡張し，Chunk タグは Start/End 法と IOB2 の両方について実験している．形態素解析の単語分割が固有表現の開始直前，および終了直後で分割されない問題には，特別な対処はしていない．訓練データは CRL 固有表現データ約 11,000 文を使用し，学習アルゴリズムは最大エントロピー法と決定リストの 2 つについて実験を行っている．最大エントロピー法を用いた場合，低頻度素性の使用による過学習を回避するため，内元らと同様の制約を用いてあらかじめ素性選択を行っている．彼らの手法の最良の結果は，Start/End タグを使用し最大エントロピー法により学習した場合で，IREX 本試験 GENERAL データに対して F 値で 82.8 と報告している．

我々は，現在本試験データを使用できないため，**表 10** に本手法における交差検定の各検定での精度を記載した．結果は最高の精度を得た前語 2 単語の単語自身，品詞細分類，文字種を素性とし，Chunk タグは IOB2，二次の多項式 Kernel 関数で左向き解析を行った場合である．本手法の平均の訓練データサイズは約 8,800 文である．比較対象とした 2 つの研究とは同一データでないため完全な比較は不可能だが，本手法の最低精度は 81.6 であり，内元らの手法と同等以上の精度が期待できる．また颯々野らの手法は固有表現の長さにより可変長文脈を考慮しているため，一部の結果で本手法を上回っている．今後 SVM を用いて学習する場合でも可変長文脈を考慮する必要がある．

比較した 2 つの手法は，過学習を回避するために，あらかじめ制約により低頻度の素性を排除し，全体の素性数を制限している．低頻度の素性を削除することは，訓練データに対する精度を犠牲にしテストデータの精度を上げることを意味する．内元らの手法は訓練データに対して F 値で約 85 と報告しており，低頻度素性を削除して過学習を回避することで訓練データの精度を犠牲にしていることが分かる．一方，本手法では素性数を減少させる制約は使用していない．本手法で訓練データに対する精度を調査した結果，交差検定におけるすべての検定において F 値で約 99 であり，訓練データに対してほぼ完全に固有表現を抽出できることが分かった．よって SVM を用いた本手法は，テストデータに対しても比較した 2 つの手法と同等以上の精度が期待でき，かつ訓練データに対する精度も犠牲にしないことから，より有効な手法であるといえる．

また比較した 2 つの手法が採用した最大エントロピー法は，素性の組合せを考慮するために，人手により組合せを定義する必要がある．しかし網羅的な組合せの定義は，計算量の面でも現実的ではなく，素性数の増加による過学習の危険性も問題となる．本手法で採用した SVM は，多項式 Kernel 関数の適用により，素性の組合せを網羅的に考慮した学習が計算量を変えることなく実現可能であり，次元数の増加による過学習の危険性は SVM の素性空間の次元数に依存しない汎化能力により軽減できる．

6. まとめと今後の課題

本稿では，IREX 日本語固有表現抽出タスクに対し，Support Vector Machine を用いて固有表現抽出規則を学習し，抽出実験においてその有効性を示した．SVM の素性空間に依存しない高い汎化能力により，単語自身，品詞細分類，文字種，およびそれらの組合せを素性として使用した高次元素性空間で，過学習することなく高い精度を得ることができた．また複数の単語からなる日本語固有表現は，その後方の単語により種類が決定される場合が多く，これに適切に対処可能な左向き解析がより良い精度を得ることが分かった．そして SVM に多項式 Kernel 関数を適用することで，素性を独立に学習するのではなく，それらの組合せを考慮した学習が精度に強く貢献することが分かった．

今後の課題としては，颯々野らの提案した可変文脈長を考慮する手法は，SVM を用いた本手法でも有効であると考えられるため，本手法に組み入れその有効性について検証することがあげられる．

謝辞 実験に使用したデータの使用を許可してくださった毎日新聞社に感謝いたします.

参 考 文 献

1) Tjong Kim Sang, E.F. and Veenstra, J.: Representing text chunks, *Proc. EACL'99*, pp.173–179 (1999).
2) IREX 実行委員会：IREX ワークショップ予稿集 (1999).
3) SAIC: *Proc. 7th Message Understanding Conference* (*MUC-7*) (1998).
4) Sekine, S., Grishman, R. and Shinnou, H.: A Decision Tree Method for Finding and Classifying Names in Japanese Texts, *6th Workshop on Very Large Corpora*, pp.171–178 (1998).
5) Kudoh, T. and Matsumoto, Y.: Use of Support Vector Learning for Chunk Identification, *Computational Natural Language Learning* (*CoNLL-2000*), pp.142–144 (2000).
6) Joachims, T.: Transductive Inference for Text Classification Using Support Vector Machines, *Machine Learning Proc. 16th International Conference* (*ICML '99*), pp.200–209 (1998).
7) KreBel, U.H.-G.: *Advances in Kernel Methods*, Chapter Pairwise Classification and Support Vector Machines, MIT Press (1999).
8) Vapnik, V.N.: *Statistical Learning Theory*, A Wiley-Interscience Publication (1998).
9) Yang, Y. and Liu, X.: A Re-examination of Text Categorization Methods, *SIGIR '99 Proc. 22nd International Conference on Research and Development in Information Retrieval*, pp.42–49, University of California, Berkeley (1999).
10) 宇津呂武仁，颯々野学：ブートストラップによる低人手コスト日本語固有表現抽出，情報処理学会研究報告，No.2000-NL-139, pp.9–16 (2000).
11) 黒橋禎夫，長尾 眞：日本語形態素解析システム JUMAN 使用説明書 version 3.6，京都大学大学院情報学研究科.
12) 松本裕治，北内 啓，山下達雄，平野善隆，松田寛，浅原正幸：日本語形態素解析システム「茶筌」version 2.0 使用説明書第二版 (1999).
13) 内元清貴，馬 青，村田真樹，小作浩美，内山将夫，井佐原均：最大エントロピーモデルと書き換え規則に基づく固有表現抽出，自然言語処理，Vol.7, No.2, pp.63–90 (2000).
14) 平 博順，春野雅彦：Support Vector Machine によるテキスト分類における属性選択，情報処理学会論文誌，Vol.41, No.4, pp.1113–1123 (2000).
15) 颯々野学，宇津呂武仁：統計的日本語固有表現抽出における固有表現まとめ上げ手法とその評価，情報処理学会研究報告，No.2000-NL-139, pp.1–8 (2000).
16) 颯々野学，斉藤由香梨，松井くにお：アプリケーションのための日本語形態素解析システム，言語処理学会第 3 回年次大会論文集，pp.441–444 (1997).

(平成 13 年 4 月 5 日受付)
(平成 13 年 11 月 14 日採録)

山田 寛康

1974 年生．1997 年山梨大学工学部電子情報工学科卒業．1999 年同大学大学院博士前期課程修了．同年奈良先端科学技術大学院大学博士後期課程入学．自然言語処理，情報検索，機械学習の研究に従事.

工藤 拓（学生会員）

1976 年生．1999 年京都大学工学部電気電子工学科卒業．2001 年奈良先端科学技術大学院大学博士前期課程修了．同年同大学院博士後期課程入学．以来，機械学習，統計的自然言語処理，情報検索の研究に従事.

松本 裕治（正会員）

1955 年生．1977 年京都大学工学部情報工学科卒業．1979 年同大学大学院工学研究科修士課程情報工学専攻修了．同年電子技術総合研究所入所．1984〜1985 年英国インペリアルカレッジ客員研究員．1985〜1987 年（財）新世代コンピュータ技術開発機構に出向．京都大学助教授を経て，1993 年より奈良先端科学技術大学院大学教授，現在に至る．工学博士．専門は自然言語処理．人工知能学会，日本ソフトウェア科学会，言語処理学会，認知科学会，AAAI，ACL，ACM 各会員.