社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－NL－179 (3)
2007－SLP－ 66 (3)
2007／5／24

# Shift-Reduce法に基づく日本語固有表現抽出

山田 寛康
(株) ジャストシステム イノベーティブ・テクノロジー研究開発部
hiroyasu_yamada@justsystem.co.jp

**概要**

本稿では日本語固有表現に対してShift-Reduce法に基づく抽出法を提案しIREX日本語固有表現抽出タスクを用いてその有効性を検証する．提案手法はShift-Reduce法に基づくことで，文頭から順に固有表現の語境界推定後にその種類を推定するという自然な解析が実現できる．また日本語における形態素単位解析では，形態素語境界と固有表現の語境界が異なる場合の誤抽出が問題となる．この問題に対し，提案手法は簡単な拡張アクションを追加することで，入力文全てを文字単位に解析することなく対処できる．CRL固有表現抽出データを用いた五分割交差検定による評価実験では，文頭から文末に向かって部分的に文字単位解析する効率的な方法で，0.88のF値を得た．

# Shift-Reduce Chunking for Japanese Named Entity Extraction

Hiroyasu Yamada
Innovative Technology R&D Dept., Justsystems
hiroyasu_yamada@justsystem.co.jp

**Abstract**

We propose a method for Japanese Named Entity(NE) extraction based on shift-reduce parsing in a deterministic manner. After *shift* action is employed to determine the word boundaries of an NE composed of multiple morphemes, *reduce* action is applied for the estimation of the NE type. In analysis of Japanese NEs for each morpheme, incorrect extractions are inevitable because of some NEs whose word boundaries are different from the morpheme's ones. While most well known work analyzes NEs for each character in sentences at the expense of efficiency, our method can analyze NEs for each morpheme in most cases by introducing two types of additional shift-reduce actions that adjust to the word boundaries of an NE. The result of 5-fold cross validation using CRL NE data-set shows that the 0.88 F-value is comparable with related work, and our left-to-right analysis for each morpheme is more efficient.

## 1 はじめに

人名・組織名といった固有表現を自動抽出する固有表現抽出問題は，情報検索，情報抽出の前処理として必須の技術であるため重要である．近年様々な分野や新しい固有表現に柔軟に対応するために機械学習を用いて抽出規則を学習する手法が多数提案されてる．特に汎化性能の高いSupport Vector Machines(SVMs)[9]を用いた手法[8, 11, 14]は決定的な解析にもかかわらず高い精度が報告されている．従来手法の多くが固有表現を固有表現の開始終了位置及びその種類を表す記号列に符号化し，抽出問題はその復号化問題として扱うことで実現している．しかし限られた周辺文脈情報から，固有表現の開始終了位置及びその種類まで同時に推定することは困難である．そこで本稿ではShift-Reduce(SR)法に基づく日本語固有表現抽出方法を提案する．SR法に基づくことで，固有表現の開始終了位置の推定後に，その種類を推定することが解析手法として自然に実現され，固有表現の種類推定に有用な周辺文脈情報を自然に扱うことが可能となる．また日本語では形態素と語境界が一致しない固有表現が存在し，形態素単位で解析する場合に誤抽出が生じる．そのため高精度が報告されている従来研究では，文字単位に固有表現を解析し，形態素境界に依存しない汎用的な手法を採用している．しかし入力文全てについて文字単位の解析を行うことは直感的にも不自然であり，また計算量の観点からいっても効率の良い方法とはいえない．提案手法では，SR法に簡単な拡張アクションを追加することで，必要な部分のみを文字単位に解析する効率的解析が実現できる．以下本稿では次章で日本語固有表現抽出タスクと従来手法及びその問題点について概説する．3章で提案するSR法に基づく手法について述べ，4章でSRアクション適用規則の学習について説明する．5章ではCRL日本語固有表現抽出データを使用した評価実験結果と関連研究との比較を報告する．最後に6章でまとめと今後の課題について述べる．

## 2 日本語固有表現抽出

表 1: IREX で定義された固有表現の種類と例

| 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 |

IREX 日本語固有表現タスク [3] では表 1 に示す 8 種類の固有表現を定義している．従来の研究の多くが形態素列が固有表現の一部か否かを識別する Chunking 問題とみなし, IOB1, IOB2, IOE1, IOE2[2], 及び SE[6] の 5 種類の符号化方法を使って最適符号列を推定する問題に帰着させている．

図 1 は"小泉首相は九日午前零時に訪朝し" という文中で, 小泉:人名 (PERSON), 九日:日付 (DATE), 午前零時:時間 (TIME), 及び朝: 地名 (LOCATION) という 4 つの固有表現に対して, IOB1, IOB2, IOE1, IOE2, SE それぞれの 表現法の違いを表している (詳細は文献参照)．ここで B-DATE のような表記を**固有表現タグ**と呼ぶこととする．また入力文に対し固有表現タグで符号化し, 任意の学習手法を適用後最適符号列に復号化する固有表現抽出手法を, 本手法と区別するために総称し**復号化手法** と呼ぶこととする．近年では 復号化手法に対し HMM-perceptron, HMM-SVMs[10], 条件付き確率場 (CRFs)[5] など, 符号列全体を考慮した様々な最適化手法が提案されている．CRFs 及びその拡張である Semi-Markov CRFs を日本語固有表現抽出に適用した研究には福岡の研究 [15] があり, 高い精度が報告されている．

### 復号化手法の問題点

復号化手法の問題の一つは, 固有表現の開始終了位置でその種類を同時に表す固有表現タグを推定しなければならないことにある．以下の例文を用いて説明する．例文で "/" は形態素境界だとする．(1) の例文で「一太郎２００７」は固有物名で, (2) の「２００７」は日付表現として抽出すべき固有表現とする．

(1) 一太郎/２/０/０/**７**/と/記載
(2) 賞味/期限/２/０/０/**７**/と/記載

いま前後 2 形態素の情報を素性として利用し文末から順[1]に解析して「７」の位置での固有表現タグ推定問題を考える．この場合二例文で利用できる素性は完全に同一なため, 固有物名か日付表現か正しく判別できない．中野ら [14] は文節境界を事前に推定し, その情報を用いることで「一太郎」や「期限」などの推定に重要な情報を利用している．しかしこれは「７」の位置で固有表現の境界とその種類を同時推定しなければならない復号化手法の本質的な問題といえる．また形態素の語境界と異なる境界の固有表現に対し, 従来法は文字単位で解析をすることで誤抽出に対対処してきた．しかし入力文全てを文字単位解析することは明らかに非効率であり, 形態素という意味のある単位で解析できるのが望ましい．本稿ではこの 2 つの問題を解決するために SR 法を拡張する．

[1]従来研究より主辞を認識できる文末から順に解析する手法が精度が高いため

## 3 SR 法に基づく日本語固有表現抽出

Shift-Reduce(SR) 法は上昇型句構造構文解析手法の 1 つであり, 文頭から順に句構造木を構築する．SR 法は, 枝が非交差で親が一意に決定される任意の木構造に対して適用可能であり, その原理は句構造解析 [1] 以外にも, 依存構造解析 [4, 12], 修辞構造解析 [7] など様々な問題に広く適用されている．固有表現抽出は, 固有表現自身を特別な句とみなせば, 固有表現か否かを句とする句構造解析と等価であり, SR 法が適用可能である．本稿では日本語固有表現抽出に適した形で利用すること, 及び SR アクションの適用規則を機械学習するために変更を加える．

先ずアルゴリズム説明に必要な用語について述べる．アルゴリズム実現にあったって便宜的に 3 種類の変数 *Left-Context*(*LC*), *Stack*[2], 及び *RightContext*(*RC*) を使用する．*LC* は解析が済んだ形態素を順に格納するための変数で, *Stack* は推定する固有表現を格納する変数である．*RC* は未解析の形態素列を保持する変数である．従って解析の初期状態は, *LC* が空の状態で, *Stack* は, 入力文の先頭の形態素を一つ読み込んだ状態, そして *RC* には残りの形態素列が格納される．この 3 つの変数に対して *Shift*, *Reduce*, 及び拡張アクションである *Cut* を適用し決定的に解析を行う．拡張 SR アクションの動作及び適用条件を表 2 に示す．各アクションの動作や適用条件は従来の SR 法と若干異なる．日本語固有表現の多くは名詞句であり, その主辞情報が抽出の重要な手掛かりとなることが従来研究の調査でわかっている．そこで従来の SR 法の *Shift* アクションのように, 複数の構成要素からなる句を認識するために *RC* から要素を追加読み込みする用途では使用しない．固有表現の最後尾要素が *Stack* に格納されてはじめて *Shift* アクションを適用し *LC* から構成形態素を一つ読み戻すという方法に変更する．*Reduce* は従来の SR 法と同一である．拡張アクション *Cut* は, 形態素語境界と異なる境界の固有表現に対し *Stack* の先頭 (*Cut-Left*) または最後尾文字 (*Cut-Right*) をポップして固有表現の語境界に一致さ

[2]拡張によりデータ構造の Stack と挙動が異なるがあえてこの用語を使用した

| 入力文 | 小泉 | 首相 | は | 九 | 日 | 午前 | 零 | 時 | に | 訪 | 朝 | し |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| タグの種類 | タグの種類による各単語の表現 | | | | | | | | | | | |
| IOB1 | I-PERSON | O | O | I-DATE | I-DATE | B-TIME | I-TIME | I-TIME | O | O | I-LOC | O |
| IOB2 | B-PERSON | O | O | B-DATE | I-DATE | B-TIME | I-TIME | I-TIME | O | O | B-LOC | O |
| IOE1 | I-PERSON | O | O | I-DATE | E-DATE | I-TIME | I-TIME | I-TIME | O | O | I-LOC | O |
| IOE2 | E-PERSON | O | O | I-DATE | E-DATE | I-TIME | I-TIME | E-TIME | O | O | E-LOC | O |
| SE | S-PERSON | O | O | B-DATE | E-DATE | B-TIME | I-TIME | E-TIME | O | O | S-LOC | O |

図 1: 固有表現に対する符号化方法の違い

せるアクションである. この二つのアクションを追加するだけで, 入力文全てを文字単位に解析することなく, 任意の長さの固有表現が解析可能となる.

解析過程例を図 2 に示す. 図 2 において最初の列は説明の便宜上付与した Step 番号である. 2 列目の *Action* は適用される SR アクションを表す. 3 列目以降が解析過程での *LC*, *Stack*, 及び *RC* の変化を表す. Step 1 では *Stack* 内が人名の固有表現 PERSON と完全に一致したため, *Reduce* をアクションを適用する. この時に固有表現の種類である PERSON を開始終了文字位置のインデックス 0-1 と共に記録しておく. 非固有表現形態素が Stack 内に格納された場合は *Reduce*-**Others** を適用し解析を進める (Step 2-3,7,10). Step 4 で「九日」の「九」が *Stack* にあるときに *Shift* をするのではなく, 主辞である「日」が *Stack* に格納されたときに *Shift* して (Step 5), *LC* から一形態素 *Stack* に読み込む. 「訪朝」の「朝」は形態素境界と固有表現境界が異なるため *Cut-Left* により境界を固有表現の境界に合わせる (Step 8-9). *Stack* 及び *RC* が空の状態になれば解析終了となる (Step 11).

# 4 SR アクションの学習

固有表現抽出アルゴリズムの疑似コードを図 3 に示す. 入力文は形態素解析器により形態素列 $(m_1, m_2, \cdots, m_n)$ に変換される. 図 3 で *get_features* は周辺文脈から素性を抽出する関数で, その素性ベクトルを $\mathbf{x}$ で表す (素性の詳細は 4.1 節で述べる). *estimate_action* で, 訓練時 **model** は訓練データに付与された正解固有表現情報を用いて, 表 2 の適用条件に従い正しい SR アクション $y$ を返す. この時の素性ベクトルとアクションのペア $(\mathbf{x}, y)$ が 1 つの訓練事例となり任意の学習アルゴリズムによりモデルを構築する. テスト時は学習したモデルが **model** となり周辺文脈素性ベクトル $\mathbf{x}$ から適切なアクション $y$ を推定する. その後 *apply_action* により, 3 種類の変数を $y$ に応じて適切に変化させ解析を進める.

**Input Sentence:** $(m_1, m_2, \cdots, m_n)$
**Initialize**:
  $LC := \{\}$;
  $Stack := \{m_1\}$;
  $RC := \{m_2, m_3, \cdots, m_n\}$;
**while** ($Stack$.size $> 0$ || $RC$.size $> 0$)
  $\mathbf{x} := get\_features(LC, Stack, RC)$
  $y := estimate\_action(\mathbf{model}, \mathbf{x})$
  $apply\_action(y, LC, Stack, RC)$
**end;**

図 3: 固有表現抽出アルゴリズム

## 4.1 学習に用いる素性

学習に使用する素性は, 従来研究で標準的に使用されている周辺文脈 (解析位置から前後 $n$ 形態素. 本稿では $n = 2$) に含まれる表層文字列, 品詞細分類情報[3], 及び文字種を基本素性とした (図 4). 提案手法の場合 *LC* の最後尾から $n$ 形態素, *Stack* 内全ての形態素, 及び *RC* の先頭要素から $n$ 形態素が周辺文脈に該当する. *Stack* 内の情報は適切なアクションを推定するために特殊な情報を使用する. 先ず *Stack* には複数の形態素列が格納され得るため, その状態を表現すために基本素性で位置を表す部分に SE 法と同じの表記を用いた[4]. さらに図 5 に示した *Stack* 内の特別な素性を使用する. *Cut* アクションの推定のため, 1 または 2 文字の接頭・接尾文字列を素性として使用した. 名詞句以外の固有表現のために *Stack* 内最右内容語及びその品詞を素性[5]として区別して使用した. また形態素境界と異なる境界をもつ固有表現のために境界情報を使用する (一致していれば BE, 先頭が不一致なら IE, 末尾が不一致なら BI). さらに特殊な固有表現は文字列全体を考慮する必要があるため *Stack* 内の文字列を 1 つにした全体文字列素性を使用した.

## 4.2 解析時の Reduce アクション

テストデータ解析時は未知の文脈に対しアクションを推定するため, 訓練時では起り得ないアクションが推定さ

[3] 上位三階層までを使用した.
[4] *Stack* 内が 5 形態素以上の場合は素性ベクトル肥大化をさけるため, *Stack* 内は先頭及び末尾 2 形態素のみを基本素性として使用した.
[5] 内容語は動詞, 名詞, 未知語, 形容詞とした

表 2: 日本語固有表現抽出用 SR アクション

| アクション | | 動作 | 適用条件 |
|---|---|---|---|
| *Reduce* | *label* | *Stack* 先頭及び最後尾文字位置と *label* を記録し，各形態素を順に *LC* の最後尾にプッシュする．*Stack* を空にした後 *RC* 先頭形態素をポップし *Stack* にプッシュする | *Stack* 内が過不足なく一つの固有表現を構成する全ての形態素列を格納した時 (*label* はその固有表現の種類)．非固有表現及び固有表現の最後尾要素が *Stack* 内にない場合は *label* を *Others* とする |
| *Shift* | | *LC* の最後尾要素から 1 形態素をポップし *Stack* の先頭にプッシュする． | 固有表現の最後尾要素と *Stack* の最後尾要素は一致するが固有表現を構成する形態素が *LC* に存在する場合 |
| *Cut* | *Left* | *Stack* の先頭 1 文字をポップし，*LC* の末尾にプッシュ | 固有表現の最後尾要素が *Stack* の最後尾要素と一致してるが，先頭要素が形態素境界と一致しない場合 |
| | *Right* | *Stack* の末尾 1 文字をポップし，*RC* の先頭にプッシュ | 固有表現の最後尾要素が *Stack* 内に含まれてるが形態素境界と一致しない場合 |

| PERSON | | | DATE | | | LOC. | | |
|---|---|---|---|---|---|---|---|---|
| 小 泉 | 首 相 | が | 九 | 日 | に | 訪 | 朝 | し | ... |

| | *Actions* | *LC* | *Stack* | *RC* |
|---|---|---|---|---|
| 1 | *Reduce-* **PERSON** | {} | { 小泉 } | { 首相, が, 九, 日, に, 訪朝, し } |
| 2 | *Reduce-* **OTHERS** | 0-1:PERSON { 小泉 } | { 首相 } | { が, 九, 日, に, 訪朝, し } |
| 3 | *Reduce-* **OTHERS** | 0-1:PERSON { 小泉, 首相 } | { が } | { 九, 日, に, 訪朝, し } |
| 4 | *Reduce-* **OTHERS** | 0-1:PERSON { 小泉, 首相, が } | { 九 } | { 日, に, 訪朝, し } |
| 5 | *Shift* | 0-1:PERSON { 小泉, 首相, が, 九 } | { 日 } | {, に, 訪朝, し } |
| 6 | *Reduce-* **DATE** | 0-1:PERSON { 小泉, 首相, が } | { 九, 日 } | { に, 訪朝, し } |
| 7 | *Reduce-* **OTHERS** | 0-1:PERSON, 5-6:DATE { 小泉, 首相, が, 九, 日 } | { に } | { 訪朝, し } |
| 8 | *Cut-* *Left* | 0-1:PERSON, 5-6:DATE { 小泉, 首相, が, 九, 日, に } | { 訪朝 } | { し } |
| 9 | *Reduce-* **LOCATION** | 0-1:PERSON, 5-6:DATE { 小泉, 首相, が, 九, 日, に, 訪 } | { 朝 } | { し } |
| 10 | *Reduce-* **OTHERS** | 0-1:PERSON, 5-6:DATE, 9-9:LOCATION { 小泉, 首相, が, 九, 日, に, 訪朝 } | { し } | {} |
| 11 | | 0-1:PERSON, 5-6:DATE, 9-9:LOCATION { 小泉, 首相, が, 九, 日, に, 訪朝, し } | {} | {} |

図 2: SR 法による解析過程例

図 4: 基本素性の例

| | 素性 | | |
|---|---|---|---|
| 位置 | 形態素 | 品詞 | 文字種 |
| -2 | 日本 | 名詞 | 漢字 |
| -1 | 五輪 | 名詞 | 漢字 |
| B | 特別 | 名詞 | 漢字 |
| I | 審査 | 名詞 | 漢字 |
| I | 委員 | 名詞 | 漢字 |
| E | 会 | 名詞 | 漢字 |
| +1 | で | 助詞 | 平仮名 |
| +2 | , | 記号 | 記号 |

図 5: *Stack* 内素性の例

| | *Stack* 内形態素 | |
|---|---|---|
| | { 特別, 審査, 委員, 会 } | |
| 境界情報 | BE | BE |
| 接頭文字 | 特, 特別 | |
| 接尾文字 | | 員会, 会 |
| 全文字列 | 特別審査委員会 | |
| 最右内容語 | | 会, 名詞 |

れる場合があり，これに対処する必要がある．文頭から順に固有表現を抽出していくと，後方の推定結果がそれ以前の推定結果と矛盾する場合がある．例えば下の例文で

中国/大陸/を/横断/する

「中国」が *Stack* にある場合この時点で LOCATION で *Reduce* したとする．その後，「大陸」の位置で *Shift* を推定し「中国大陸」が *Stack* に格納され再び LOCATION で *Reduce* した場合，「中国」のみが LOCATION の固有表現という以前の推定結果と矛盾する．直感的にはより有用な文脈情報を利用できる後方の結果を優先するほうが自然である．本稿では，以前の推定結果を保守する場合と

後方の推定結果を優先し決定的ではあるが自身の推定結果を訂正していく方法との違いを実験で比較する.

## 5 実験

**データと実験設定**: 実験では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, 10,718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. また形態素解析器は茶筌[13]を使用した. CRL固有表現データを記事単位で5分割した交差検定を行い, テストデータに対する評価は $\beta = 1$ の $F$ 値を使用した. 学習アルゴリズムはSVMsを使用し, 多値分類問題はOne vs. rest法を, Kernel関数としては二次の多項式 $(\mathbf{x}_1 \cdot \mathbf{x}_2 + 1)^2$ を用いた.

### 5.1 結果

表3にSR法による抽出精度を示す. 表3で, **前向き** *Shift* は, *Shift* 動作を従来のSR法同様に固有表現の開始位置から順にStack最後尾にプッシュする方法で学習及び解析した場合の精度を示す. **前方優先**は前方で推定した結果が後方の推定結果と矛盾した場合に前方を優先し, 推定結果の訂正を行わない場合の精度を表す. **後方優先**は逆に後方の推定結果を優先して前方の推定結果を訂正した場合の精度を表す. **文節素性**は中野らが提案した文節素性[14][6]と同等な素性を追加して学習し後方優先で解析した場合の精度を示す.

表3: 実験結果

| | 平均 $F_{\beta=1}$ 値 | | | |
|---|---|---|---|---|
| 固有表現の種類 | 前向き *Shift* | 前方優先 | 後方優先 | 文節素性 |
| ORGANIZATION | 80.57 | 81.05 | 83.25 | **84.53** |
| PERSON | 87.57 | 86.99 | 88.61 | **88.79** |
| LOCATION | 88.50 | 87.73 | 89.35 | **89.59** |
| ARTIFACT | 50.92 | 51.10 | 53.78 | **54.49** |
| DATE | 90.67 | 92.14 | 93.37 | **93.89** |
| TIME | 89.52 | 89.02 | 89.15 | **90.43** |
| MONEY | 60.11 | 91.02 | **94.73** | 93.88 |
| PERCENT | 89.01 | 92.09 | 94.65 | **96.42** |
| 総合 | 85.42 | 86.13 | 87.79 | **88.33** |

SR法元来の *Shift* アクションと同じ**前向き** *Shift* は, 主辞を認識してから逆向きに *Shift* する他の3手法に比べ精度が低い. この結果は主辞要素の認識が固有表現の推定に重要であるという過去の研究結果の知見と一致する. また中野らの提案した文節素性は提案手法においてもMONEYを除く全ての固有表現において F 値向上させており, 固有表現か非固有表現かの推定に貢献していると考えられる.

[6]文献[14]に報告されている model B と同等. 文節境界推定にはCaboCha[12]を使用した.

表5: 解析計算コストの比較

| 手法 | 事例 | ラベル | #SVs | 素性 | L1 |
|---|---|---|---|---|---|
| 文字 | 99,626 | 17 | 77,194 | 237,014 | 23.5 |
| SR | 65,532 | 12 | 67,522 | 168,735 | 24.6 |
| 比率 | 65.8% | 70.6% | 87.5 % | 71.2% | 104.7% |

### 5.2 関連研究との比較

本手法の有効性を検証するため, 近年高い精度を報告したいくつかの関連研究との比較を行った. 表4に比較結果と各手法の特徴概要を示す.

文節素性の使用有無で精度の変化はあるが, 提案手法はSVMsをつかった文字単位復号化手法である浅原ら中野らの手法[7]と同等の精度を達成している. 従来研究が文末から解析する必要があるのに対し, 提案手法は文頭から解析しても主辞を認識して解析できる. より直感的で自然な解析順を保持しても精度が低下しないことも提案手法の特徴といえる.

#### 解析時における計算コストの比較

従来の復号化手法と提案手法の解析時における計算コストを比較する. 先ず復号化手法では, 推定すべきラベル数は固有表現の種類数を $c$ とすると, IOB及びIOE符号化で $2c+1$ 個のラベル数が必要となる. これに対して本手法の最大ラベル数は $c+4$ となる. 従って固有表現の種類が4種類以上であれば必ず少ないラベル数の推定問題となる. 齋藤ら[16]は評判抽出タスクの前処理にIREXに準拠した固有表現をさらに17種類に細分化している. このような関連研究を考慮すれば, SR法の少ない推定ラベル数は今後より効率的な解析への貢献が期待できる.

SVMsを用いて高精度を達成した中野らや浅原らの手法は, 固有表現境界が形態素境界と異なる問題に対し, 文字単位解析で対応している. この場合解析時も文字数がそのまま分類事例数となる. これに対して, 提案手法では適用したSRアクション数が事例数となる. 表5に交差検定の1つテストセットを解析するのに必要となった事例数, 訓練データの素性数, Support Vector 数, 推定ラベル数, 及び素性ベクトルの平均L1ノルムの比較をまとめた. 文字単位解析の数値は中野らの model B をから算出した.

提案手法の事例数は文字単位復号化手法の約3分の2に縮小しており, 入力文すべてを文字単位で解析しない部分が効率化に寄与している. 素性数においても中野らが用いた素性は, 全ての周辺文脈に形態素境界情報を付随させて素性展開しているため結果的に提案手法よりも高次元の素性空間になっている. 学習された SVs の数も提案

[7]浅原らの冗長解析結果2次まで使用した場合と比較. 両研究の意味情報を利用しない場合の最良結果と比較.

表 4: 従来研究との比較

| | 総合 F 値 | 解析方向 | 解析単位 | 学習手法 | その他 |
|---|---|---|---|---|---|
| 提案手法 | 87.79 | 文頭 → 文末 | 部分的に文字 | SVMs | 文節素性無 |
| | 88.33 | 文頭 → 文末 | 部分的に文字 | SVMs | 文節素性有 |
| 浅原 [8] | 86.35 | 文末 → 文頭 | 文字 | SVMs | 形態素冗長解析結果使用 |
| 福岡 [15] | 87.71 | - | 文字 | semi-CRF | 文節境界素性 |
| | 87.81 | - | 文字 | linear-CR | 文節境界素性 |
| 中野ら [14] | 87.07 | 文末 → 文頭 | 文字 | SVMs | 文節素性無 |
| | 88.78 | 文末 → 文頭 | 文字 | SVMs | 文節素性有 |

手法が少ない. 復号化手法は推定ラベル数が多いため, 必要となる SVs も結果的に多くなった. 素性数は工夫次第で削減可能であり, また SVs の数は学習アルゴリズム固有の問題であるため一般的な言及はできないが, 事例数とラベル数では復号化手法よりも少なく個々の素性ベクトルの長さはほぼ同じであるため, 同等程度の精度実現する場合, 提案手法がより効率的だといえる.

### (Semi-Markov) CRFs との比較

福岡ら及び齋藤らが適用した CRFs は固有表現タグ系列全体を考慮して最適化するため, 解析方向には依存せず, 周辺文脈長にも影響をうけにくく適切な学習結果が期待できる. また福岡らが適用した Semi-Markov CRFs は固有表現の境界も含め系列全体を最適化する学習手法であり, 固有表現境界と形態素境界の不一致を直接学習アルゴリズムで解決できる期待がある. しかし今回使用した CRL 固有表現データを使用した実験結果からは SVMs を用いた決定的な解析手法と同等以下の結果になっており期待通り精度改善効果は得られていない.

## 6 まとめと今後の課題

本稿では, SR 法に基づく日本語固有表現抽出法を提案し, IREX 固有表現抽出タスクを用いてその有効性を検証した. SR 法の適用により, 固有表現の境界推定後その種類を推定する自然な解析手法が実現できた. また形態素境界と固有表現境界が一致しない問題も, 簡単な拡張アクションの追加で対応可能で, 任意の長さの固有表現を解析できることを示した. 実験の結果, 従来の復号化手法と比べ推定ラベル数が少なくより簡潔な推定問題になっていること, 及び部分文字単位解析でも同程度の精度を実現できたことから, より効率的な解析手法であることが分かった.

今後の課題は, より頑健な固有表現抽出を実現するために, 低コストで学習データを拡充できる能動学習を用いる予定である. さらに今回提案した手法をこれまで復号化問題で高精度を得た他の Chunking 問題に適用する予定である.

**謝辞** 論文執筆にあたり, 情報提供及び有用な助言をして下さった奈良先端科学技術大学院大学 浅原正幸氏 に感謝致します.

## 参考文献

[1] Adwait Ratnaparkhi. Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning*, Vol. 34, No. 1-3, pp. 151–175, 1999.

[2] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 173–179, 1999.

[3] IREX 実行委員会 (編). IREX ワークショップ予稿集, 1999.

[4] Joakim Nivre and Jens Nilsson. Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 99–106, 2005.

[5] John Lafferty and Andrew McCallum and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *The International Conference on Machine Learning (ICML)*, pp. 282–289, 2001.

[6] Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. Named Entity Extractin Based on A Maximum Entropy Model and Transformation Rules (in Japanese). In *Journal of Natural Language Processing*, Vol. 7, pp. 63–90, 2000.

[7] Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, 2000.

[8] Masayuki Asahara and Yuji Matsumoto. Japanese Named Entity Extraction with Redundant Morphological Analysis. In *HLT-NAACL*, pp. 8–15, 2003.

[9] Vladimir N. Vapnik. *Statistical Learning Theory*. A Wiley-Interscience Publication, 1998.

[10] Y. Altun and I. Tsochantaridis and T. Hofmann. Hidden Markov Support Vector Machines. In *The International Conference on Machine Learning (ICML)*, 2003.

[11] 磯崎秀樹, 賀沢秀人. 固有表現抽出のためのＳＶＭの高速化. 情報処理学会論文誌, Vol. 44, No. 3, pp. 970–979, 2003.

[12] 工藤 拓、松本 裕治. チャンキングの段階適用による日本語係り受け解析. Vol. 43, No. 6, pp. 1834–1842, 2002.

[13] 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, 12 1999.

[14] 中野桂吾, 平井有三. 日本語固有表現抽出における文節情報の利用. 情報処理学会論文誌, Vol. 45, No. 3, pp. 934–941, 2004.

[15] 福岡 健太. Semi-Markov Conditional Random Fields を用いた固有表現抽出に関する研究. 奈良先端科学技術大学院大学情報科学研究科修士論文, 2003.

[16] 齋藤邦子, 鈴木潤, 今村賢治. CRF を用いたブログからの固有表現抽出. 言語処理学会第 13 回年次大会, pp. 107–110, 2007.