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

2005－NL－170（22）
2005／11／22

機械学習を用いた日本語複合辞のチャンキング

注連隆夫[†1]　内元清貴[†2]　土屋雅稔[†3]
高木俊宏[†4]　宇津呂武仁[†1]
佐藤理史[†5]　井佐原均[†2]

本稿では，Support Vector Machine(SVM) を用いたチャンカー YamCha を利用して，日本語複合辞の検出器を学習し，その性能評価を行った．SVM は従来からある学習モデルと比較して，入力次元数に依存しない高い汎化能力を持ち，Kernel 関数を導入することで効率良く素性の組合わせを考慮しながら分類問題を学習することが可能である．SVM を日本語複合辞の検出に適用し，実際のタグ付きデータを用いて解析を行った結果，日本語複合辞を構成している形態素の数の情報，形態素の日本語複合辞における位置情報を考慮した場合，交差検定により F 値で約 94 という高精度の結果が得られた．

Chunking Japanese Compound Functional Expressions by Machine Learning

Takao SHIME ,[†1] Kiyotaka UCHIMOTO ,[†2]
Masatoshi TSUCHIYA ,[†3] Toshihiro TAKAGI ,[†4]
Takehito UTSURO ,[†1] Satoshi SATO [†5]
and Hitoshi ISAHARA [†2]

This paper proposes to learn a detector of Japanese compound functional expressions using the chunker YamCha based on Support Vector Machines (SVMs), and presents the result of evaluating the performance of the detector. It is well-known that SVMs achieve high generalization performance, and furthermore, by introducing the Kernel principle, SVMs can carry out training with smaller computational cost independent of the dimensionality of the feature space. As features of SVM learning, we consider the number of morphemes constituting a compound functional expression, and the position of each morpheme within a functional expression. We achieve the cross validation result of the F-value as 94, which shows the effectiveness the proposed method.

1. はじめに

複合辞とは，幾つかの語が複合してひとまとまりの形となって非構成的な意味を持ち辞的な機能をはたす表現である[4]．この複合辞に対して，それと同一表記をとり，構成的な意味をもつ表現が存在する．例えば，「にあたって」という表現は，「出発するにあたって，荷物をチェックした」という文では，「する時に」に相当する複合辞であるが，「太郎は，壁にあたって怪我をした」という文では構成的な意味として扱われる．このため，このような複合辞は，構成的な意味で扱われる表現と区別する必要がある．

しかし，既存の解析系はいずれも，そのような処理を全ての複合辞に適用しきれていない．例えば，形態素解析器 JUMAN[15] と構文解析器 KNP[16] の組合わせは，形態素解析時には複合辞を検出しない．構文解析時に，解析規則に記述された特定の形態素列が現れると，直前の文節の一部としてまとめたり，直前の文節からの係り受けのみを受けるように制約を加えて解析を行うといった，複合辞を意識した処理を行う．我々が日本語複合辞用例データベース[7] において対象としている複合辞のうち，区別

†1 京都大学大学院 情報学研究科
Graduate School of Informatics, Kyoto University
†2 情報通信研究機構
National Institute of Information and Communications Technology
†3 豊橋技術科大学 情報メディア基盤センター
Information and Media Center, Toyohasi University of Technology
†4 京都大学 工学部
Kyoto University, Faculty of Engineering
†5 名古屋大学大学院 工学研究科
Graduate Schoole of Engineering, Nagoya University

して処理する必要がある複合辞は，少なくとも 111 種類あるが，JUMAN/KNP では 21 種類（約 19%）しか区別されていない．

次に，形態素解析器 ChaSen[11] と構文解析器 CaboCha[9] の組合わせを利用して，構文解析を行う場合を考える．その際，形態素解析器は IPA 品詞体系（THiMCO97）の形態素解析用辞書[12] を用い，構文解析器は，京都テキストコーパス[17] から機械学習したモデルを用いるとする．この場合，形態素解析用辞書に「助詞・格助詞・連語」と登録されている複合辞は，形態素解析時に検出される．また,「ざるを得ない」などの表現は直前の文節の一部としてまとめられる．区別して処理しなければならない 111 種類の複合辞のうち，この組合わせでは 14 種類（約 13%）しか区別されない．

日本語複合辞の体系的な検出システムの構築を目的として，人手で作成した規則を用いて複合辞を検出する手法が提案されてきた[3),5),8)]．しかし，これらの手法では複合辞検出規則を人手で作成するのに多大なコストがかかっていた．

そこで本稿では，SVM を用いたチャンカー YamCha[14] を利用した日本語複合辞の検出器を提案する．日本語複合辞用例データベース[7] を訓練データとして学習した日本語複合辞検出器によって，土屋らが提案した人手による手法[8] と比べ，複合辞を高精度に検出できることを示す．

2. SVM を用いたチャンカー YamCha

YamCha は Support Vector Machine(SVM) を用いたチャンク解析器である．この節では，SVM について，および YamCha を用いた学習・解析方法について述べる．

2.1 Support Vector Machine

サポートベクトルマシンとは，空間を超平面で分割することにより 2 つのクラスからなるデータを分類する二値分類器のことである．2 つのクラスを正例，負例とすると，学習データにおける正例と負例の間隔 (マージン) を最大にする超平面を求め，それを用いて分類を行う．通常は，学習データにおいてマージンの内部領域に少数の事例が含まれてもよいとする拡張 (ソフトマージン) や，超平面の線形の部分を非線形とする拡張 (カーネル関数の導入) などがなされたものが用いられる．これらの拡張によりクラスを判別することは，以下の識別関数の出力値が正か負かによってクラスを判別することと等価である[1),2)]．

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (1)$$

$$b = -\frac{max_{i,y_i=-1} b_i + min_{i,y_i=1} b_i}{2}$$

$$b_i = \sum_{j=1}^{l} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i)$$

ここで $\mathbf{x}$ は識別したい事例の文脈 (素性の集合) を，$\mathbf{x}_i$ と $y_i (i = 1, ..., l, y_i \in \{1, -1\})$ は学習データの文脈とクラスを意味する．また，関数 $sgn(x)$ は，$x \geq 0$ のときに 1，$x < 0$ のときに -1 となる二値関数であり，各 α_i は式 (3) と式 (4) の制約のもと式 (2) の $L(\alpha)$ を最大にするものである．

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \quad (2)$$

$$0 \leq \alpha_i \leq C \ (i = 1, ..., l) \quad (3)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (4)$$

また，関数 K はカーネル関数と呼ばれ，様々なものが提案されているが，本論文では次の式で表される多項式カーネルを用いる．

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d \quad (5)$$

ここで，C, d は実験的に設定される定数である．本論文では C,d はそれぞれ 1 と 2 に固定した．

サポートベクトルマシンは二値分類器であるため，クラスの数が 2 であるデータしか扱えないが，これにペアワイズ手法を組み合わせることにより，クラスの数が 3 以上のデータを扱えるようになる．

ペアワイズ手法とは，N 個のクラスを持つデータの場合，異なる 2 つのクラスのあらゆるペア (N(N-1)/2 個) を作り，各ペアごとにどちらがよいかをサポートベクトルマシンなどの二値分類器で求め，最終的に N(N-1)/2 個の二値分類器のクラスの多数決により，最適なクラスを求める方法である．

2.2 YamCha での学習・解析方法

YamCha は，形態素単位，単語単位，文節単位など様々な単位のチャンク同定問題に対応することができる．本研究では，形態素単位のチャンク同定問題を扱う．この節では，実際に我々が行った学習・解析方法について説明する．

チャンクの学習に用いるチャンクタグは以下で示すような IOB2 フォーマット[6] のものが使われることが多い．本研究でもこの IOB2 フォーマットを使用している．

- **I** 複合辞の途中にある形態素
- **O** 複合辞でない形態素
- **B** 複合辞の先頭の形態素

また，YamCha のパラメータを以下のように設定し，学習・解析を行った．

- 学習解析方向
  - 前向き
- カーネル関数
  - 2 次の多項式カーネル
- 多値分類への拡張手法
  - ペアワイズ法
- 学習・解析に用いる素性

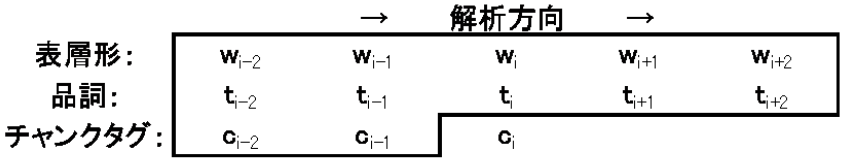

図 1 YamCha の学習・解析

表 1 判定ラベル体系

判定ラベル	判定単位	読み	内容 vs 機能	用法	複合辞
B	不適切				—
Y	適切	不一致			×
C	適切	一致	内容的	内容的用法	×
F	適切	一致	機能的	用例集で説明されている用法	◎
A	適切	一致	機能的	接続詞的用法	○
M	適切	一致	機能的	その他の機能的用法	○ or ×

- 今，学習（解析）している形態素の情報 (w_i, t_i)
- 前２つの形態素の情報 $(w_{i-2}, w_{i-1}, t_{i-2}, t_{i-1})$ とチャンクタグ (c_{i-2}, c_{i-2})
- 後ろ２つの形態素の情報 (w_{i+1}, w_{i+2})

図 1 での c_i の学習・解析に用いる素性は，$w_{i-2}, \cdots, w_{i+2}, t_{i-2}, \cdots, t_{i+2}, c_{i-2}, c_{i-1}$ となる．そして，これらの素性をベクトル空間における要素と考え SVM を用いて学習・解析を行う．用いる素性のうち，前２つの形態素のチャンクタグ (c_{i-2}, c_{i-1}) は学習時には与えられているが，解析開始時には，与えられていない．これらの素性は解析時に逐次，自動的に追加されていく．

3. YamCha を用いた複合辞検出器

3.1 日本語複合辞用例データベースにおける判定ラベル

本研究では，日本語複合辞用例データベースで定義されている判定ラベルを用いて，チャンクタグを作成している．判定ラベルとは，複合辞候補 (複合辞として用いられている可能性がある表現) が文中でどのような働きをしているかを表すラベルである．判定ラベルの体系を**表 1** に示す．

判定ラベル B は，複合辞候補が判定単位として不適切であることを表す．例えば，文 (1) の複合辞候補「上で」は，名詞「屋上」の一部と助詞「で」からなっており，判定単位としては不適切であるから，文 (1) の複合辞候補には判定ラベル B を付与する．

(1) ビルの屋 上で 犬が鳴いている．

判定ラベル Y は，複合辞候補の読みが，判定対象となっている複合辞の読みと一致しないことを表す．例えば，判定対象の複合辞「うえで」の用例として文 (2) が得られた場合，その複合辞候補「上で」の読みは「じょうで」であり，複合辞の読み「うえで」と一致していない．この複合辞候補には，判定ラベルとして Y を付与する．

(2) 地球 上で 何かが起こっている．

表 2 チャンクタグ体系

チャンクタグ体系１ (CHK1)	F	A	M	C	Y	B
チャンクタグ体系２ (CHK2)	F	A	M	C	Y	B
チャンクタグ体系３ (CHK3)	F	A	M	C	Y	B
チャンクタグ体系４ (CHK4)	F	A	M	C	Y	B
チャンクタグ体系５ (CHK5)	F	A	M	C	Y	B

判定ラベル C は，複合辞候補に内容的に働いている語が含まれていることを表す．例えば，文 (3) の複合辞候補「と言っても」に含まれている動詞「言う」は本来の意味で内容的に働いているので，この複合辞候補に判定ラベルとして C を付与する．

(3) 彼に「勉強をしろ」と言っても 無駄だ．

判定ラベル F，A，M は，複合辞候補が機能的に働いているとき，その機能を区別するためのラベルである．判定ラベル F は，複合辞候補が現代語複合辞用例集[10] で説明されている用法で働いていることを表す．判定ラベル A は，複合辞候補が接続詞的用法で用いられていることを表す．判定ラベル M は，複合辞候補がこれら以外の機能的な働きをしていることを表す．例えば，複合辞候補「ところで」の用例として，文 (4)〜(6) に存在する複合辞候補を判定する場合を考える．

(4) 倍率が上がった ところで，入学金があがることはない．

(5) ところで，彼が今月結婚したことを知っていますか．

(6) 彼は，あと一歩の ところで １位をとることができなかった．

文 (4) の複合辞候補「ところで」は現代語複合辞用例集で説明されている通りの働きをしているので，判定ラベルとして F を付与する．文 (5) の複合辞候補「ところで」は文頭にあって接続詞的に働いているので，判定ラベルとして A を付与する．文 (6) の複合辞候補「ところで」は，形式的にはたらいている名詞「ところ」を含んでいるので，判定ラベルとして M を付与する．

3.2 チャンクタグ体系

本研究では，上で述べた判定ラベルのうち，F ラベルが付与される表現（複合辞）を検出する検出器 F と，F, A，M のいずれかのラベルが付与される表現（機能表現）を検出する検出器 FAM を提案する．これらを実現するために**表 2** のような 5 種類のチャンクタグ体系を用意した．チャンクタグ体系 1 (CHK1) は，判定ラベル F，A，M，C，Y，B を全て区別してチャンクタグを付与する．チャンクタグ体系 2 (CHK2) は，F，A，M ラベルを 1 つのチャンクとし，それら以外のラベルは，ラベル別にチャンクタグを付与する．チャンクタグ体系 3 (CHK3) は，F, A，M ラベルと C，Y，B ラベルをそれぞれ 1 つのチャンクとしている．チャンクタグ体系 4 (CHK4) は，F ラベルとそれ以外のラベルの 2 つのチャンクを用いる．チャンクタグ体系 5 (CHK5) は，F ラベル，A，M ラベル，C，Y，B ラベルをそれぞれ 1 つのチャンクとしている．

検出器 F は CHK1，CHK4，CHK5 を，検出器 FAM は CHK1，CHK2，CHK3，CHK5 を用いた．

3.3 学習・解析に用いる素性

本研究で提案する検出器は，学習・解析の際，以下の 3 種類の情報を素性として用いる．

- 形態素解析器 MeCab[13] を利用して，IPA 品詞体系での形態素解析を行うことによって得られる形態素の情報，以下 10 種類 (これらを S10 とする)
  - 表層形，品詞，品詞細分類 1 ～ 3，活用型，活用形，原形，読み，発音
- 複合辞候補を構成している形態素の数（複合辞候補の長さ）と複合辞候補における形態素の位置の情報 (これらを S2 とする)
- 複合辞候補の後ろに続く 2 つの形態素の素性集合 S10 と S2 の情報 (これらを S24 とする)

ただし，S24 は，検出したい全ての表現を 1 つの学習器で学習する際は，悪影響を与えるので，表現ごとに学習を行うときにのみ使用する．

以降，学習・解析の際，素性集合 S10 と S2 をあわせて使う場合を S12，素性集合 S10 と S2 と S24 をあわせて使う場合を S36 と記述する．

複合辞候補「上で」を含む文 (7) を例として考える．

(7) 十分に検討した 上で，慎重に判断したい．

この場合は，**図 2** のように全ての形態素に，形態素解析によって得られる情報を付与し，さらに複合辞候補「上で」を構成している各形態素に，複合辞候補を構成している形態素数である 2 という値を，複合辞候補を構成している「上」,「で」の形態素に位置情報である 1，2 という値を付与する．

表層形	し	た	上	で	、	慎重
品詞	動詞	助動詞	名詞	助詞	記号	名詞
品詞細分類1	自立	*	非自立	格助詞	読点	形容動詞語幹
品詞細分類2	*	*	副詞可能	一般	*	*
品詞細分類3	*	*	*	*	*	*
活用型	サ変・スル	特殊・タ	*	*	*	*
活用形	連用形	基本形	*	*	*	*
原形	する	た	上	で	、	慎重
読み	シ	タ	ウエ	デ	、	シンチョウ
発音	シ	タ	ウエ	デ	、	シンチョー
形態素の位置	*	*	1	2	*	*
複合辞候補の長さ	*	*	2	2	*	*
チャンクタグ	O	O	B	I	O	O

図 2 学習・解析に使用する素性

4. 実験と考察

本稿で提案する 2 つ検出器，検出器 F と検出器 FAM の 2 つの検出器に対して，性能評価実験を行った．また，土屋らの提案するルールベースの複合辞検出器[8] と本稿で提案する検出器との比較実験を行った．最後に，学習データの量による性能比較実験を行った．

4.1 実験環境，設定

実験には，日本語複合辞用例データベースを使用した．このデータベースは複合辞 304 表現に対し，毎日新聞 (1995 年)[18] から 10,968 用例を抽出して，複合辞候補に判定ラベルを付与したものである．このうち，本研究では，判定ラベル F とそれ以外の用法がバランスよく収録されている 51 表現に対する 2550 用例 (1 つの表現について 50 用例) を基に作成したデータを使用した．この時，下に述べる 2 種類の作業を行った．

2 つの複合辞が共通する文字列を含む場合，日本語複合辞用例データベースの判定ラベルをそのまま用いることができない．例えば，複合辞「といっても」の用例収集時に文字列照合により複合辞「ても」の用例が収集された場合，日本語複合辞用例データベースでは C ラベルが付与される．しかし，この用例は，複合辞「ても」の用例収集時に文字列照合により収集された場合は，F ラベルが付与されるはずである．本研究では，各複合辞ごとに学習を行っているのではなく，複合辞全体に対して 1 つの学習を行っているので，この場合,「といっても」に C ラベルを付与するのではなく,「ても」に F ラベルを付与する方が適切である．それゆえ，我々は，日本語複合辞用例データベースに存在する全てのこのような場合に対して，上のような修正作業を行った．この修正データセットをデータセット a と呼ぶ（判定ラベルの数は**表 8** のデータセット 3 の判定ラベルの数に等しい)．

日本語複合辞用例データベースの用例においては，対象である 1 つの複合辞候補にしか判定ラベルが付与されていない (判定ラベルの数は表 8 のデータセット 1 の判定ラベルの数に等しい)．しかし，実際には 1 つの用例に対象外の複合辞候補が含まれることが多く，これに対して判定ラベルを与えなくては，学習するのに都合が悪い．それゆえ，我々は，用例に含まれる全ての複合辞候補に判定ラベルを付与する作業を行った．

上記 2 種類の作業を行って得られたデータセットをデータセット b と呼ぶ．

4.2 評価尺度

実験の評価には，精度，再現率，F 値を用いた．それらの算出方法は以下に示す通りである．

表 3 検出器 F の条件の違いによる検出性能

	CHK1		CHK4		CHK5	
	S10	S12	S10	S12	S10	S12
精度	0.849	0.901	0.845	0.905	0.851	0.905
再現率	0.899	0.942	0.884	0.933	0.892	0.938
F 値	0.873	**0.921**	0.865	0.918	0.871	**0.921**

$$\text{精度} = \frac{\text{検出に成功したチャンク数}}{\text{解析によって検出されたチャンク数}}$$

$$\text{再現率} = \frac{\text{検出に成功したチャンク数}}{\text{評価データに存在するチャンク数}}$$

$$\text{F 値} = \frac{2 \times \text{再現率} \times \text{精度}}{\text{再現率} + \text{精度}}$$

実験データを 10 等分し，10 分割交差検定を行い，10 回の評価結果の平均値を示す．

4.3 検出器 F，FAM の評価実験

チャンクタグの体系 (CHK1，CHK4，CHK5) と使用する素性 (S10，S12) の組み合わせによる 6 つの条件で比較実験を行った．精度や再現率を求める際，注目するチャンクは判定ラベルが F のチャンクである．実験結果を**表 3** に示す．

チャンクタグの体系 (CHK1，CHK2，CHK3，CHK5) と使用する素性 (S10，S12) の組み合わせによる 8 つの条件で比較実験を行った．精度や再現率を求める際，注目するチャンクは判定ラベルが F，A，M のチャンクである．ただし，F，A，M の差を無視して評価を行った．つまり，たとえ F のチャンクを A のチャンクとして検出しても検出成功とみなした．実験結果を**表 4** に示す．

表 3，表 4 より，全てのチャンクタグ体系において，S12 の条件は，S10 の条件に比べ，精度，再現率，F 値全ての数値が約 5 ポイント上昇しているのがわかる．これは，複合辞候補を構成している形態素数の情報と複合辞候補を構成している形態素の位置情報の 2 つの情報 (S2) が，表現を検出するのに効果的な素性であることを示している．また，チャンクタグ体系の違いによる性能の差は，ほとんど見られなかった．これは，複合辞を検出するためには，CHK4 で，機能表現を検出するためには，CHK3 で学習を行えば十分であることを示している．

4.4 人手により作成した規則を用いた検出器との性能比較

検出器 F，検出器 FAM と，土屋らが提案したルールベースの複合辞検出器[8](F ラベルが付与される表現を検出するものを検出器 F-R，F，A，M ラベルが付与される表現を検出するものを検出器 FAM-R と呼ぶ) との性能比較実験を行った．検出器 F は，4.3 節の実験で最も性能のよかった条件 (CHK5,S12) で，検出器 FAM は，実験で最も性能のよかった条件 (CHK5,S12) で検出を行った．評価は，表現全体での評価と表現ごとの評価の二通り行った．実験結果を**表 5** に示す．

表 5 から，検出器 F，FAM の方が，検出器 F-R，FAM-R に比べ，精度，再現率，F 値全ての数値が高いことが見て取れる．これにより，機械学習による複合辞検出の優位性を示すことができた．

さらに，検出器 FAM，FAM-R の検出性能を表現ごとに測定した結果を**表 6** に示す． この結果から，平均して検出器 FAM の方が検出器 FAM-R より F 値で約 5 ポイント性能がよいのがわかる．しかし，全ての表現において，検出器 FAM が検出器 FAM-R より検出性能がよいわけではない．

表現ごとに見てみると，検出器 FAM-R の方が検出器 FAM よりも検出性能がよい表現が 51 表現中 11 表現存在した．そのうち F 値が 5 ポイント以上差がついた表現は，A05-2000，A24-1000，A37-1000，A37-2000，A42-1000，A51-2000 の 6 表現であった．

A05-2000(といえば)，A37-2000(にあたり)，A42-1000(に応じて) の 3 表現は，前後の形態素の情報だけでは，機能表現を検出するのが難しい用例が多く，学習が困難である．その結果，検出器 FAM では，精度，再現率がともに低い数値となり F 値も低い数値となった．それに対し，検出器 FAM-R では，再現率重視のルールが採用されているので，再現率がとても高い数値となり，結果的に F 値も高くなった．

A24-1000(おりから)，A37-1000(にあたって)，A51-2000(にしたがい) の 3 表現は，検出器 FAM-R では，高い精度で検出できている．しかし，検出器 FAM では，検出器 FAM-R に比べ検出性能が大幅に落ちる．この現象は，検出器 FAM では，51 表現全てを 1 つの学習器で学習しており，各表現に特化した性能を発揮することができないということが原因であるとと考えられる．そこで，この 3 つの表現に対して，表現ごとに検出器の学習を行った後，性能を評価した．この時，学習の条件は (CHK5,S36) である．その結果を**表 7** に示す．この表を見ると，全ての表現において検出性能は向上しており，検出器 FAM-R と比べても遜色のないものとなっている．これより，これらの表現では，表現ごとに検出器を学習するのが望ましいことがわかる．

次に，検出器 FAM でも検出器 FAM-R でも検出性能が悪く，F 値が 0.8 未満であった A04-2000(とすると)，A09-6000(とおもったら)，A37-2000(にあたり)，および，先ほどの考察で前後の形態素の情報だけでは，検出が難しいと判明した A05-2000(といえば)，A42-1000(に応じて) において，検出に利用できる係り受け情報がないか調べた．その結果，A37-2000 以外の表現は，検出に有効な情報を見つけることができなかった．A37-2000 において解析に失敗したのは，名詞が前接している 18 用例であり，そのうち F ラベルのものが 10 用例，C ラベルのものが 8 用例であった．解析に失敗した用例の係り受け情報を調査したところ，C ラベルでは，8 用例中 6 用例で，主語が複合辞候補を含む文節に係っているのに対し，F ラベルでは，主語は複合辞候補を含む文節には係っていなかった．よって，主語が複合辞候補を含む文節に係るか

表 4 検出器 FAM の条件の違いによる検出性能

	CHK1		CHK2		CHK3		CHK5	
	S10	S12	S10	S12	S10	S12	S10	S12
精度	0.880	0.925	0.883	0.921	0.876	0.926	0.880	0.929
再現率	0.911	0.956	0.923	0.957	0.909	0.956	0.910	0.955
F 値	0.895	0.940	0.902	0.939	0.892	0.941	0.894	**0.942**

表 5 検出器の性能比較

検出器	表現全体での評価			表現ごとの評価		
	精度	再現率	F 値	精度	再現率	F 値
検出器 F	**0.905**	**0.938**	**0.921**	**0.892**	**0.916**	**0.902**
検出器 F-R	0.874	0.766	0.816	0.877	0.790	0.803
検出器 FAM	**0.929**	**0.955**	**0.942**	**0.906**	**0.928**	**0.916**
検出器 FAM-R	0.876	0.886	0.881	0.874	0.890	0.867

どうかという情報は，機能表現を検出するのに有効な情報であると考えられる．

4.5 学習データの量による性能比較

一つの用例に複数の複合辞候補が存在することがあるので，データセット b においては，各複合辞に対するチャンクタグの数は一様ではない．しかしながら，用例として文ではなく，チャンクタグを学習するのに必要な最小限の形態素列を利用して，適切に学習が行えるのであれば，学習データの量を各複合辞ごとに均一にすることができ，さらに，学習データを作成するコストも削減することができる．一方，最小限の形態素列のみを用いて学習を行った場合，O タグ (複合辞候補を構成していない形態素のチャンクタグ) の学習が不十分となり，O タグの検出性能が著しく悪くなる．しかし，本研究では，F ラベルが付与される表現や F，A，M のいずれかが付与される表現と，それらと同一表記をとる C，Y，B ラベルのいずれかが付与される表現を区別することに焦点を当て，複合辞候補以外の形態素列に対してはチャンクタグ判定を行わないので，この問題は無視できる．

そこで，本実験では，検出器 F，検出器 FAM において，(CHK5,S12) の条件で，以下で示す 3 つのデータセットで学習後，複合辞の検出を行い，最小限の形態素列を用いての学習と文を用いての学習の間における性能差について調べた．

- データセット 1
  - データセット a のラベル付与部分から，複合辞候補とその前後 2 つずつの形態素を切り出したデータ．
- データセット 2
  - データセット a のラベル付与部分から，複合辞候補とその前後 2 つずつの形態素，さらに，複合辞候補の前後 2 つずつの形態素に他の複合辞候補の一部が含まれていた場合，その複合辞も含め，前後 2 つずつの形態素に複合辞候補が存在しなくなるまで形態素列を拡張して切り出したデータ．
- データセット 3
  - データセット b に対して，データセット 2 と同じ処理を行い，形態素列を切り出したデータ．

データセット 1 とデータセット 2 の差分は複合辞候補が接近して現れた時に利用できる情報の量であり，データセット 2 とデータセット 3 の差分は学習データの量である．各データセットにおける複合辞候補の数，複合辞候補が接近して出現している箇所の数を表 8 に示す．

実験結果を**表 9** に示す．実験結果から，データセット 1，2，3 の F 値の差は最大でも 1 ポイントであることがわかる．データセット 1 の結果とデータセット 2 の結果にほとんど差がないことは，複合辞候補が接近して現れた時に利用できる情報を無視したとしても，学習に支障をきたさないということを示している．また，データセット 1 の結果とデータセット 3 の結果にほとんど差がないことは，最小限の形態素列を利用して学習を行っても検出性能を維持できることを示している．よって，作成コストが少ないデータセット 1 を用いても適切な学習が行うことが可能であることがわかった．

5. おわりに

本稿では，SVM を用いたチャンカー YamCha を利用して，日本語複合辞の検出器を学習し，その性能評価を行った．この性能評価を通して，機械学習による複合辞の検出は，人手による規則を用いた複合辞の検出よりも高い性能を発揮することを示すことができた．

また，複合辞候補とその前後 2 つずつの形態素に限って学習を行った場合でも，この制約を課さない場合とほとんど性能が変わらないことから，学習データを作成するコストを大幅に削減することが可能であることがわかった．

今後の研究課題としては，検出器の検出対象の複合辞の種類を増やし，その性能を評価することが挙げられる．また，表現ごとに学習を行ったところ，F 値が 2 ポイント以上改善したものが 11 表現，0 以上 2 未満改善したものが 11 表現，変らなかったものが 6 表現，下がったものが 23 表現であった．この情報を基に，表現ごとの検出器学習が必要な複合辞と，不必要な複合辞の区別を行い，最適な性能でかつコンパクトな検出器を構成する計画である．

表 6 検出器 FAM，FAM-R の表現ごとの検出性能比較（詳細）

ID	表記	精度		再現率		F 値	
		検出器 FAM	検出器 FAM-R	検出器 FAM	検出器 FAM-R	検出器 FAM	検出器 FAM-R
A01-2000	といっても	0.892	0.842	0.917	0.914	0.904	0.877
A01-3000	とはいっても	1.000	0.979	1.000	0.959	1.000	0.969
A02-1000	とはいえ	1.000	0.947	0.974	1.000	0.987	0.973
A04-1000	とすれば	0.977	0.946	1.000	0.854	0.989	0.897
A04-2000	とすると	0.722	0.909	0.813	0.625	0.765	0.741
A04-3000	としたら	0.967	1.000	0.967	1.000	0.967	1.000
A05-1000	というと	0.846	0.583	0.917	0.840	0.880	0.689
A05-2000	といえば	0.829	0.810	0.763	0.919	0.795	0.861
A06-2000	となれば	0.957	0.933	1.000	0.955	0.978	0.944
A07-1000	というものの	0.960	0.893	0.960	1.000	0.960	0.943
A08-1000	といいながら	0.766	0.766	0.947	0.947	0.847	0.847
A09-1000	かと思うと	0.833	0.629	0.833	0.917	0.833	0.746
A09-6000	と思ったら	0.697	0.75	0.852	0.75	0.767	0.75
A12-1000	うえで	1.000	0.893	0.974	0.714	0.987	0.794
A12-1010	うえでの	1.000	0.875	0.974	0.525	0.987	0.656
A17-1000	くせに	0.949	0.971	1.000	0.892	0.974	0.930
A20-1000	ところを	0.909	0.865	0.882	0.914	0.896	0.889
A24-1000	**おりから**	**0.889**	**1.000**	**0.800**	**1.000**	**0.842**	**1.000**
A30-1000	ても	0.929	0.808	0.929	0.825	0.929	0.816
A33-1000	のにたいして	1.000	0.978	1.000	1.000	1.000	0.989
A33-2000	のにたいし	1.000	1.000	1.000	1.000	1.000	1.000
A34-1000	にしても	0.952	0.769	1.000	1.000	0.976	0.870
A34-3000	にしろ	1.000	0.731	1.000	0.633	1.000	0.679
A34-4000	にせよ	0.964	0.786	0.964	0.786	0.964	0.786
A35-1000	だけに	1.000	0.984	1.000	1.000	1.000	0.992
A37-1000	**にあたって**	**0.705**	**0.944**	**0.886**	**1.000**	**0.785**	**0.971**
A37-2000	にあたり	0.556	0.477	0.476	1.000	0.513	0.646
A38-1000	にあって	0.778	0.750	0.933	1.000	0.848	0.857
A42-1000	に応じて	0.837	0.974	0.923	0.974	0.878	0.974
A42-1011	に応じた	0.792	0.553	0.905	1.000	0.844	0.712
A44-1000	にかけては	0.980	1.000	1.000	0.208	0.990	0.345
A47-1011	に先だち	1.000	1.000	1.000	0.974	1.000	0.987
A51-2000	**にしたがい**	**0.875**	**1.000**	**0.700**	**0.800**	**0.778**	**0.889**
A56-2000	にとり	0.750	0.737	0.857	1.000	0.800	0.848
A59-2000	にかけ	0.893	0.952	1.000	0.909	0.943	0.930
A62-1000	として	0.929	0.876	0.958	0.832	0.944	0.853
A63-1000	としては	1.000	0.966	1.000	1.000	1.000	0.983
A70-1000	をはじめ	0.953	0.976	0.976	0.932	0.965	0.953
A82-1000	という	0.958	0.892	0.984	0.753	0.971	0.816
A82-2000	との	0.993	0.926	1.000	0.993	0.997	0.958
B04-1000	ところだ	0.865	0.909	0.938	0.698	0.900	0.789
B12-1000	ことがある	0.837	0.827	0.900	1.000	0.867	0.905
B12-1030	ことがない	0.917	0.942	0.957	1.000	0.936	0.970
B12-3000	こともある	0.981	0.909	0.945	0.980	0.963	0.943
B12-3030	こともない	0.979	0.980	0.959	0.980	0.969	0.980
B12-4000	ことはない	0.872	0.849	0.932	0.957	0.901	0.900
B18-1000	ばかりだ	1.000	0.926	1.000	1.000	1.000	0.962
B19-1000	にきまっている	0.882	1.000	0.750	0.700	0.811	0.824
B29-7000	てはならない	0.974	0.976	0.949	0.930	0.961	0.952
B30-2000	ていい	0.875	0.625	0.921	1.000	0.897	0.769
B33-1000	てならない	1.000	1.000	1.000	1.000	1.000	1.000
表現ごとの平均		0.906	0.874	0.928	0.890	0.916	0.867

参 考 文 献

1) Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
2) Taku Kudoh. TinySVM: Support Vector Machines. `http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/index.html`, 2000.
3) 松吉俊, 佐藤理史, 宇津呂武仁. 接続情報にもとづく助詞型機能表現の自動検出. 言語処理学会第 11 回年次大会論文集, pp. 1044–1047, 2005.
4) 森田良行, 松木正恵. 日本語表現文型, NAFL 選書, 第 5 巻. アルク, 1989.
5) 中塚裕之, 佐藤理史, 宇津呂武仁. 助動詞型機能表現

表 7 表現ごとの学習による検出性能

ID	精度		再現率		F 値	
	検出器 FAM	検出器 FAM-R	検出器 FAM	検出器 FAM-R	検出器 FAM	検出器 FAM-R
A24-1000	0.909	1.000	1.000	1.000	0.952	1.000
A37-1000	1.000	0.944	1.000	1.000	1.000	0.971
A51-2000	1.000	1.000	0.800	0.800	0.889	0.889

表 8 データセット

データセット	判定ラベル							複合辞候補が接近して出現している箇所
	F	A	M	C	Y	B	合計	
データセット 1	1483	52	310	453	8	154	2460	0
データセット 2	1537	52	332	465	8	157	2551	85
データセット 3	2039	55	449	525	9	171	3248	89

表 9 データセットの違いによる検出性能

データセット	検出器 F			検出器 FAM		
	精度	再現率	F 値	精度	再現率	F 値
データセット 1	0.897	0.925	0.911	0.925	0.945	0.935
データセット 2	0.902	0.934	0.918	0.929	0.952	0.941
データセット 3	0.905	0.938	0.921	0.929	0.955	0.942

の形態・接続情報と自動検出. 言語処理学会第 11 回年次大会論文集, pp. 596–599, 2005.

6) E. Tjong Kim Sang. Noun phrase recognition by system combination. In *Proceedings of the 1st Conference of NAACL*, pp. 50–55, 2000.

7) 土屋雅稔, 松吉俊, 宇津呂武仁, 佐藤理史, 中川聖一. 日本語機能表現用例コーパスの作成. 言語処理学会第 11 回年次大会論文集, pp. 986–989, 2005.

8) 土屋雅稔, 宇津呂武仁, 佐藤理史, 中川聖一. 形態素情報を用いた日本語機能表現の検出. 言語処理学会第 11 回年次大会論文集, pp. 584–587, 2005.

9) 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. 情報処理学会論文誌, 第 43 巻, pp. 1834–1842, 2002.

10) 国立国語研究所. 現代語複合辞用例集, 2001.

11) 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸. 形態素解析システム「茶筌」version 2.3.3 使用説明書. http://chasen.naist.jp/, 2003.

12) 浅原正幸, 松本裕治. ipadic version 2.6.1 ユーザーズマニュアル. `http://chasen.aist-nara.ac.jp/chasen/doc/ipadic-2.6.1-j.pdf`, 2003.

13) 工藤拓. 形態素解析器 MeCab. `http://chasen.org/~taku/software/mecab/`.

14) 工藤拓, 松本裕治. Support Vector Machine を用いた Chunk 同定. 自然言語処理研究会 2000-NL-140, 2000.

15) 黒橋禎夫, 河原大輔. 日本語形態素解析システム JUMAN version 5.1 使用説明書, 9 2005. `http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman/juman-5.1.tar.gz`.

16) 黒橋禎夫, 河原大輔. 日本語構文解析システム KNP version 2.0 使用説明書, 9 2005. `http://www.kc.t.u-tokyo.ac.jp/nl-resource/knp/knp-2.0.tar.gz`.

17) 黒橋禎夫, 長尾眞. 京都大学テキストコーパス・プロジェクト. 言語処理学会第 3 回年次大会発表論文集, pp. 115–118, 1997.

18) 毎日新聞社. CD-毎日新聞'95 データ集. 日外アソシエーツ, 1996. `http://www.nichigai.co.jp/sales/mainichi/mainichi-data.html`.