2007－NL－177 (9)
2007／1／26

機能表現を考慮した統計的日本語係り受け解析

注連隆夫[†1] 土屋雅稔[†2] 松吉俊[†1,†3]
宇津呂武仁[†4] 佐藤理史[†3]

本稿では，Support Vector Machine (SVM) を用いたチャンカー YamCha を利用して，日本語機能表現検出器を学習し，その性能評価を行った．機能表現を構成している形態素の数の情報，機能表現中における形態素の位置情報を素性として参照することにより，F値で約 94 という高精度の検出器を実現できることを示した．また，京都テキストコーパスに対して，機能表現の情報を人手で付与した後，SVM に基づく統計的係り受け解析器 CaboCha の学習を行い，その性能を評価した．機能表現を考慮して係り受け関係の学習をすることによって，機能表現を含む文節の係り受け解析の性能が改善することを示す．

Statistical Dependency Analysis of Japanese coping with Functional Expressions

Takao SHIME ,[†1] Masatoshi TSUCHIYA ,[†2]
Suguru MATSUYOSHI ,[†1,†3] Takehito UTSURO [†4]
and Satoshi SATO [†3]

This paper proposes to learn a detector of Japanese functional expressions using the chunker YamCha based on Support Vector Machines (SVMs), and presents the result of evaluating the performance of the detector. Through experimental evaluation, we achieve the F-measure as 94. We then manually annotate parsed sentences of Kyoto Text Corpus with functional expressions, which are used for training dependency analyzer CaboCha based on SVM. The dependency analyzer CaboCha of this paper is modified so that it can cope with annotation of functional expressions in the training corpus. We experimentally show that the modified version of the dependency analyzer improves the performance of the dependency analysis of functional expressions.

1. はじめに

機能表現とは，「にあたって」や「をめぐって」のように，2つ以上の語から構成され，全体として1つの機能的な意味をもつ表現である．一方，この機能表現に対して，それと同一表記をとり，内容的な意味をもつ表現が存在することがある．例えば，文 (1) と文 (2) には，「にあたって」という表記の表現が共通して現れている．

(1) 出発するにあたって、荷物をチェックした。
(2) ボールは、壁にあたって跳ね返った。

†1 京都大学大学院 情報学研究科
Graduate School of Informatics, Kyoto University
†2 豊橋技術科学大学 情報メディア基盤センター
Information and Media Center, Toyohashi University of Technology
†3 名古屋大学大学院 工学研究科
Graduate School of Engineering, Nagoya University
†4 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

文 (1) では，下線部はひとかたまりとなって，「機会が来たのに当面して」という機能的な意味で用いられている．それに対して，文 (2) では，下線部に含まれている動詞「あたる」は，動詞「あたる」本来の内容的な意味で用いられている．これらの表現においては，機能的に用いられている場合と，内容的に用いられている場合とを識別する必要がある．以下，文 (1)，文 (2) の下線部のように，表記のみに基づいて判断すると，機能的に用いられている可能性がある部分を**機能表現候補**と呼ぶ．

ここで，日本語複合辞用例データベース[4](以下，**用例データベース**と呼ぶ) は，機能表現の機械処理を研究するための基礎データを提供することを目的として設計・編纂されたデータベースである．この用例データベースは，現代語複合辞用例集[6] に収録されている 125 種類の複合辞および，その異形 (合計 337 種類の機能表現) を対象として，機能表現候補と一致する表記のリストと，個々の機能表現候補に対して最大 50 個の用例を毎日新聞 (1995年) から収集したものを収録している．そして，各機能表

現候補が文中において果たしている働きが，機能的用法・内容的用法のいずれであるかを人手で付与している☆．

用例データベースにおいて，機能的用法か内容的用法かの判別が必要な表現は 111 表現存在する．111 表現に対する，既存の解析系の扱いを調べてみた．形態素解析器 JUMAN[12] と構文解析器 KNP[13] の組み合わせによって，機能的な意味で用いられている場合と内容的な意味で用いられている場合とが識別される可能性がある表現は 111 表現中 43 表現である．また，形態素解析器 ChaSen[15] と構文解析器 CaboCha[11] の組合わせを用いた場合には，識別される可能性がある表現は 111 表現中 40 表現である．

このような現状を改善するには，機能表現候補の用法を正しく識別する検出器と検出器によって検出される機能表現を考慮した係り受け解析器が必要である．

そこで本論文では，SVM を用いたチャンキングによって機能表現検出器を実現し[3),7)]，その機能表現検出器と工藤らの SVM を用いた統計的係り受け解析手法[11] を利用して構築した機能表現を考慮した係り受け解析器を使用して，機能表現を考慮した係り受け解析を実現している．

本論文の構成は以下の通りである．最初に，本論文で実現した機能表現検出器について述べる (2 章)．3 章では，機能表現を考慮した係り受け解析器を構築する際に利用した SVM を用いた統計的係り受け解析手法について述べる．4 章では，機能表現を考慮した係り受け解析器の構築方法について述べる．5 章では，実際に機能表現を考慮した係り受け解析に対して，実験，評価を行い，最後に結論を述べる (6 章)．

2. 機能表現検出器

本論文で使用している機能表現検出器は，SVM を用いたチャンキングによって実現している．具体的には SVM を用いたチャンカー YamCha[10] を利用して，形態素解析器 ChaSen による形態素解析結果を入力とする機能表現検出器を実装した．本章では，機能表現検出器の実現方法について述べた後，実際に機能表現検出器に対して，実験，評価を行っている．

2.1 チャンクタグの表現方法

チャンキングを行うためには，チャンクタグを定義する必要がある．本論文では，検出対象とする機能表現全てに共通のチャンクタグを，形態素を単位として付与している．チャンクタグは，そのチャンクタグが付与された形態素が，検出対象とする機能表現のいずれかに含まれるか否かを表し，チャンクの範囲を示す要素とチャンクの用法を示す要素という 2 つの要素からなる．以下，本論文で用いたチャンクタグについて詳細を述べる．

表 1 チャンクタグ

		用法	
		機能的用法	内容的用法
機能表現候補の形態素	チャンクの先頭	B-F	B-C
	チャンクの先頭以外	I-F	I-C
それ以外の形態素		O	

チャンクの範囲を示す要素の表現法としては，以下で示すような IOB2 フォーマット[2)] が広く利用されている．本論文でも，この IOB2 フォーマットを使用する．

- **I** チャンクに含まれる形態素 (先頭以外)
- **O** チャンクに含まれない形態素
- **B** チャンクの先頭の形態素

ただし，本論部では IOB2 フォーマットを，さらに**表 1** のように機能表現候補の用法によって細分化したものを使用する．

本論文では，2 つの用法のうち，機能的用法を検出する機能表現検出器を作成する．

2.2 素　　性

学習・解析に用いる素性について説明する．文頭から i 番目の形態素 m_i に対して与えられる素性 F_i は，形態素素性 $MF(m_i)$，チャンク素性 $CF(i)$，チャンク文脈素性 $OF(i)$ の 3 つ組として，次式によって定義される．

$$F_i = \langle MF(m_i), CF(i), OF(i) \rangle$$

形態素素性 $MF(m_i)$ は，形態素解析器によって形態素 m_i に付与される情報である．本論文では，IPA 品詞体系 (THiMCO97) の形態素解析用辞書[8)] に基づいて動作する形態素解析器 ChaSen による形態素解析結果を入力としているため，以下の 10 種類の情報 (表層形，品詞，品詞細分類 1 ～ 3，活用型，活用形，原形，読み，発音) を形態素素性として用いた．

チャンク素性 $CF(i)$ とチャンク文脈素性 $OF(i)$ は，i 番目の位置に出現している機能表現候補に基づいて定まる素性である．今，下図のような形態素列 $m_j \ldots m_i \ldots m_k$ からなる機能表現候補 E が存在したとする．

$$m_{j-2} \quad m_{j-1} \quad \boxed{m_j \ldots m_i \ldots m_k} \quad m_{k+1} \quad m_{k+2}$$

機能表現候補 E

チャンク素性 $CF(i)$ は，i 番目の位置に出現している機能表現候補 E を構成している形態素の数 (機能表現候補の長さ) と，機能表現候補中における形態素 m_i の相対的位置の情報の 2 つ組である．チャンク文脈素性 $OF(i)$ は，i 番目の位置に出現している機能表現候補の直前 2 形態素および直後 2 形態素の形態素素性とチャンク素性の組である．すなわち，i 番目の位置に対する $CF(i)$ および $OF(i)$ は次式で表される．

$$CF(i) = \langle k-j+1,\ i-j+1 \rangle$$

$$OF(i) = \langle\ MF(m_{j-2}), CF(m_{j-2}), MF(m_{j-1}), CF(m_{j-1}), MF(m_{k+1}), CF(m_{k+1}), MF(m_{k+2}), CF(m_{k+2})\ \rangle$$

☆ 実際には，機能的用法については，「現代語複合辞用例集」の用法であることを表すラベル F，接続詞的用法であることを表すラベル A，その他の機能的用法であることを表すラベル M という細分類がされており，一方，内容的用法については，機能表現候補は，用法判定単位として不適切であることを表すラベル B，機能表現候補の読みが，判定対象の機能表現の読みと一致しないことを表すラベル Y，内容的用法であることを表す C という細分類がされている．本稿では，ラベル F，A，M を機能的用法として統合し，ラベル C，B，Y を内容的用法として統合して扱っている．

チャンク素性，チャンク文脈を付与する際，複数の機能表現候補が部分的に重複して現れる場合がある．その場合，それらの候補全てに基づいてチャンク素性とチャンク文脈素性を付与するという方法と，それらの候補から何らかの基準を用いて 1 つの候補を選択し，選択された候補に基づいてチャンク素性とチャンク文脈素性を付与するという方法が考えられる．前者の方法で付与された素性を参照して機械学習を行うには，重複する可能性がある機能表現の全ての組み合わせに対して十分な量の学習事例が必要であるが，そのような学習事例を準備することは現実的ではない．そのため，本論文では，後者の方法を採り，次の優先順序に従って選ばれた 1 つの機能表現候補に基づいて，チャンク素性とチャンク文脈素性を付与することにする．

1 先頭の形態素が，最も左側の機能表現候補を用いる．
2 1 を満たす候補が複数存在する場合は，その中で最も形態素数が多い候補を用いる．

例えば，文 (3) には，「なくてはいけません」および「てはいけません」という 2 つの機能表現候補が，部分的に重複して現れている．

(3) 慎重にし なくてはいけません。

この場合，「なくてはいけません」という機能表現候補が，「てはいけません」という機能表現候補に比べて，より左の形態素から始まっているので，「なくてはいけません」という機能表現候補に基づいて，チャンク素性とチャンク文脈素性を付与する．また，文 (4) には，「という」および「というものの」という 2 つの機能表現候補が，部分的に重複して現れている．

(4) それが試合 というものの 難しさだ。

この場合，2 つの機能表現候補の先頭の形態素は同一であるため，より形態素数が多い候補「というものの」に基づいて，チャンク素性とチャンク文脈素性を付与する．

i 番目の形態素に対するチャンクタグを c_i とすると，チャンクタグ c_i の学習・解析を行う場合に用いる素性として，i 番目の形態素および前後 2 形態素に付与された素性 $F_{i-2}, F_{i-1}, F_i, F_{i+1}, F_{i+2}$ と，直前 2 形態素に付与されたチャンクタグ c_{i-2}, c_{i-1} を用いる (**図 1**)．解析時には，解析によって得られたチャンクタグを，直前 2 形態素に付与されたチャンクタグとして順に利用して，解析を行う．前後 3 形態素の素性と直前 3 形態素のチャンクタグを用いて学習・解析を行う予備実験も行ったが，前後 2 形態素の素性と直前 2 形態素のチャンクタグを用いた場合に比べて，殆んど性能が変わらなかったため，前後 2 形態素の素性と直前 2 形態素のチャンクタグを用いる．

2.3 実験と考察

本論文で提案する機能表現検出器に対して，学習および解析を行い，各ベースラインと性能を比較した．

2.3.1 データセット

文を単位として学習を行うには，文中に現れる全ての機能表現候補に対して用法ラベルが付与されたデータが必要である．ただし，用法ラベルとは，機能表現候補が内容的用法で扱われているか，機能的用法で扱われているかを示すラベルを示している．本稿では，判別が必要な 111 表現のなかでも，新聞記事においても，機能的用法と内容的用法の両方が一定の割合で出現する 32 表現を対象とする (実際には，そのような表現は 60 表現程度存在するが，データ整備の都合上，本稿の範囲では 32 表現のみを対象とした)．そして，これらの 32 表現に対する用例として用例データベースに収録されている 1600 例文 (1 つの表現につき 50 例文) について，これらの例文に含まれている全ての機能表現候補に用法ラベルを付与した．これらのデータセットに含まれる各用法の数と，全形態素数を**表 2** に示す．1 つの例文に，複数の機能表現候補が出現する場合があるため，機能表現候補の総数は，例文の総数よりも多くなっている．

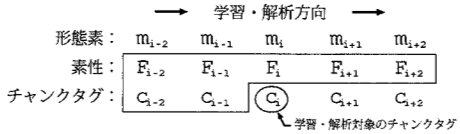


図 1 YamCha の学習・解析

表 2 データセットの各統計量

用法			全形態素数
機能的用法	内容的用法	計	
1509	721	2233	57570

2.3.2 評価尺度

実験を評価する際の尺度には，以下の式で表される精度，再現率，F 値，および判別率を用いた．

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

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

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

$$\text{判別率} = \frac{\text{正解した判定ラベル数}}{\text{全判定ラベル数}}$$

また，実験は，10 分割交差検定を用いて行った．

2.3.3 評価結果

本論文で提案する機能表現検出器と，各ベースラインの検出性能を**表 3** に示す．

表 3 において，「頻度最大の用法ラベル」とは，全ての候補部分に対して頻度最大の用法ラベル (機能的用法のラベル) を付与した場合の検出性能である．「人手作成の規則による検出器」は，土屋らが提案している手法[5] による検出性能である．

表 3 中の「CRF を用いた検出器」は，Conditional Random Fileds(CRF)[1] によって学習・解析を行った場合の検出性能である．CRF とは，系列ラベリング問題のために設計された識別モデルであり，正しい系列ラベリングを他の全ラベリング候補と弁別するような学習を行う．本論文で

表 3 機能表現検出の評価結果 (%)

		精度	再現率	F 値	判別率
ベースライン	頻度最大の用法ラベル	66.0	100	79.5	66.0
	JUMAN/KNP	76.2	33.0	46.0	53.8
	ChaSen/CaboCha	69.0	17.7	28.2	46.1
人手作成の規則による検出器		82.5	85.2	83.8	76.6
CRF を用いた検出器		91.8	93.2	92.5	89.3
SVM を用いた検出器	形態素素性	92.5	93.7	93.0	90.3
	形態素素性，チャンク素性	93.1	94.3	93.7	91.5
	形態素素性，チャンク素性，チャンク文脈素性	92.8	95.1	93.9	91.6

は，CRF による学習・解析用ツールとして CRF++☆を利用した．素性としては，前後 2 形態素の形態素素性，チャンク素性，チャンク文脈素性と，直前 2 形態素のチャンクタグを用いた．学習時には，事前分布として Gaussian Prior を用いて事後確率を最大化することにより，パラメータを正則化した[9]．その際のハイパーパラメータとしては，1,2,3,4,5 の 5 通りの値について予備実験を行い，最も良い性能を示した 1 を採用した．

表 3 中の「SVM を用いた検出器」は，本論文の提案する SVM によるチャンキング手法による検出性能である．表より，提案手法は，学習・解析に用いた素性に関わらず，ベースラインおよび人手作成の規則による検出よりも，高い F 値を示した．また，提案手法は，CRF を用いた検出器よりも，高い F 値を示した．

学習・解析に用いた素性の違いによる性能の違いを検討すると，形態素素性のみを用いた場合よりも形態素素性とチャンク素性を併用した場合の方が，形態素素性とチャンク素性を併用した場合よりも形態素素性，チャンク素性，チャンク文脈素性すべてを使用した場合の方が検出性能がすぐれていることから，チャンク素性とチャンク文脈素性は，機能表現を検出するための素性として適当であったといえる．

全ての素性を用いて学習と解析を行った機能表現検出器において，他の表現と比較して極端に検出性能が悪く，F 値が 70 に達しなかった表現は，「にあたり」の 1 表現である．例えば，文 (5) に含まれる「にあたり」は，「(新規参入という) 時が来たのに当面して」という機能的な意味で用いられているため，判定ラベル F が付与されるべき文である．それに対して，文 (6) および文 (7) に含まれる「にあたり」は，内容的に用いられているため，判定ラベル C が付与されるべき文である．

(5) 新規参入 にあたり、潜在的なニーズを掘り起こそうと、転勤族を主な対象にした。

(6) お神酒の瓶が女性 にあたり、けがをする事故があった。

(7) 米国の最先端の科学者が知恵を結集して原爆の開発 にあたり、一九四五年八月に広島・長崎に原爆が投下された。

しかし，SVM を用いた検出器では，文 (5) と文 (6) に対しては判定ラベル C を，文 (7) に対しては判定ラベル F を付与してしまい，用法を正しく判定できたのは文 (6) のみだった．仮に，文 (5) と文 (6) を区別することだけが必要ならば，直前がサ変名詞であることが有効な素性として働く可能性があるが，文 (7) は，そのような素性だけではうまく判定できない．このように，提案手法によっては適切に検出できない表現もごく少数ながら存在するが，他の表現については，表 3 に示したように適切に検出することができた．

3. SVM を用いた統計的係り受け解析

本論文で提案している機能表現を考慮した係り受け解析は，工藤らの SVM を用いた統計的係り受け解析器 CaboCha[11] を利用している．本章では，工藤らの手法の学習・解析アルゴリズム，学習・解析に使用する素性について述べる．

3.1 学習・解析アルゴリズム

工藤らの手法は，入力文 B に対する，条件付き確率 P(D|B) を最大にする係り受けパターン列 D を求める従来の手法と異なり，チャンキングを段階的適用することによって係り受け解析を実現している．ここで登場した入力文 B とは，あらかじめ文節にまとめられ，属性付けされた文節列 $b_1, b_2, ..., b_m$ を表しており，係り受けパターン列 D とは，$Dep(1), Dep(2), ..., Dep(m-1)$ を表している．ただし，$Dep(i)$ は，文節 b_i の係り先文節番号を示す．実際には，以下のようなアルゴリズムを使用して，段階的にチャンキングを行っている．

(1) 入力文節すべてに対し，係り受けが未定という意味の O タグを付与する．

(2) 文末の文節を除く O タグが付与された文節に対し，直後の文節に係るか推定．係る場合は D タグを付与．後から 2 番目の文節は無条件に D タグを付与．

(3) O タグの直後にあるすべての D タグおよびその文節を削除する．

(4) 残った文節が一つ (文末の文節) の場合は終了，それ以外は 2. に戻る．

このアルゴリズムによる解析例を**図 2** に示す．

このアルゴリズムにおける係り受け関係の同定には，SVM を用いている．この場合，従来手法では，訓練データ中の全ての 2 文節の候補を学習事例として抽出していた．しかし，このような抽出方法では，学習データを不必要に多くしてしまい，学習の効率が悪い．それに対し

☆ http://chasen.org/~taku/software/CRF++/

初期化

Input:	彼は	彼女の	暖かい	真心に	感動した。
Tag:	O	O	O	O	O
Input:	彼は	彼女の	暖かい	真心に	感動した。
Tag:	O	O	$D_{削除}$	D	O
Input:	彼は	彼女の	真心に	感動した。	
Tag:	O	$D_{削除}$	D	O	
Input:	彼は	真心に	感動した。		
Tag:	O	$D_{削除}$	O		
Input:	彼は	感動した。			
Tag:	$D_{削除}$	O			
Input:	感動した。				
Tag:	O	終了			

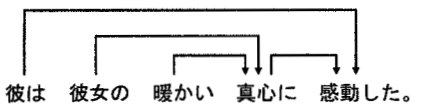


図 2 係り受け解析例

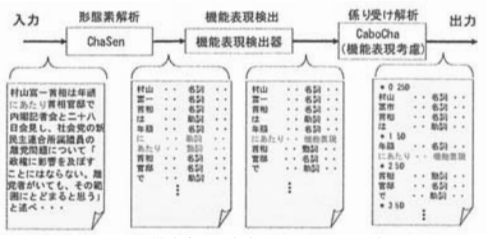


図 3 機能表現を考慮した係り受け解析

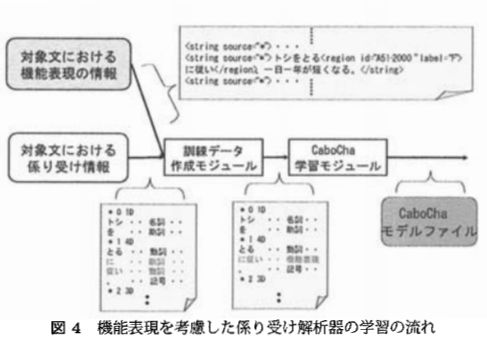


図 4 機能表現を考慮した係り受け解析器の学習の流れ

て，工藤らの手法では，学習も解析時と同じアルゴリズムを採用することにより，学習で使われる文節組のセットを，隣り合う文節組のみとしている．これにより，負例を不必要に増えるのを防ぐことができている．

3.2 学習・解析に使用する素性

SVM の学習・解析に使用する素性は，**表 4** に示す通りである．静的素性とは，文節の作成時に決定される素性を示しており，動的素性とは，係り関係そのものを素性としたものである．また，主辞とは文節内で品詞が特殊，助詞，接尾辞となるものを除き，文節末に一番近い形態素を指し，語形とは文節内で品詞が特殊となるものを除き，文節末に一番近い形態素のことを指す．

4. 機能表現を考慮した係り受け解析

本論文で提案している機能表現を考慮した係り受け解析の流れは，**図 3** の通りである．まず，ChaSen によって形態素解析を行う．次に，形態素解析結果に対して，機能表現検出器を用いて，機能表現検出を行う．その際，検出された機能表現は，構成している形態素列を連結し，一つの形態素として出力される．最後に，その出力結果に対して，機能表現を考慮した係り受け解析器を用いて，係り受け解析を行う．

本論文では，SVM を用いた統計的係り受け解析手法の学習・解析ツールとして CaboCha を利用して機能表現を考慮した係り受け解析器を実現している．具体的には，CaboCha の係り受け解析における訓練データを機能表現を考慮したものに変換するという方法を用いている．機能表現を考慮した係り受け解析の訓練データを作成するために必要な情報は二つある．一つは，対象文における係り受け解析の訓練データ．もう一つは，対象文における機能表現の情報．この二つの情報を用いて，**図 4** の流れで訓練データを作成し，学習を行っている．

図 4 の訓練データ作成モジュールでは，末尾の文節から順番に以下のアルゴリズムに従って処理を行っている．

1. 機能表現を構成している形態素列を連結する．
2. 連結する形態素列が複数の文節にまたがっている場合，文節の連結も行う．連結後の文節の係り先は，連結文節中の末尾の文節の係り先を採用する．次のステップでは，助詞・助動詞型の機能表現は，3a へ，接続詞型の機能表現は 3b へ進む．

3a. 連結した文節の先頭形態素が，機能表現の場合は，直前の文節に連結する．連結後の文節の係り先は，連結文節中の末尾の文節の係り先を採用する．

3b. 機能表現のみで一文節を構成するように文節を分解する．

4. 文節の連結，分解に伴う文節 ID，係り先の変化を反映させる．

機能表現を考慮した係り受け解析器の学習において，形態素を連結して作られた機能表現に対して，新たに品詞名を付与する必要がある．用例データベースによると，機能表現は，接続詞相当の働きをするもの (接続詞型) と助詞相当の働きをするもの (助詞型)，助動詞相当の働きをするもの (助動詞型) に分類することができる．さらに，助詞型の機能表現は，接続助詞相当のもの (接続辞類)，格助詞相当のもの (連用辞類)，連体助詞相当のもの (連体辞類) に細分類することができる．そこで，本論文では，**表 5** のような品詞体系を採用した．そして，現代語複合辞用例集[6] に掲載されている各機能表現と品詞分類との

表 4 係り受けの学習・解析に使う素性

静的素性	係り元/係り先の文節	主辞見出し，主辞品詞，主辞書品詞細分類，主辞活用，主辞活用形，語形見出し，語形品詞，語形品詞細分類，語形活用，語形活用形，括弧有無，句読点の有無，文節の位置 (文頭，文末)
	文節間	距離 (1,2-5,6 以上), 助詞，括弧，句読点の有無
動的素性	係り先に係る文節の静的素性	
	係り元に係る文節の静的素性	
	係り元が係る文節の静的素性	

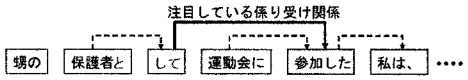


図 5 係り受け解析例 (機能表現考慮せず)

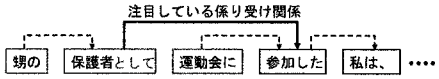


図 6 係り受け解析例 (機能表現考慮)

対応に基づいて，機能表現への品詞の付与を行った．特に，接続詞型になる可能性のある機能表現については，文頭に出現した場合は接続詞型とし，文頭以外の場合は助詞型とした．

機能表現を考慮した係り受け解析の学習と機能表現を考慮しない係り受け解析の学習における学習に使用する素性の変化を**図 5**，**図 6** の文における「して」,「保護者として」という文節と「参加した」という文節の係り受け関係の学習・解析に使用する素性について見てみる．まず，図 6 においては，文節の区切りが機能表現を考慮したものになっている．そして，それによって注目する文節が図 5 では,「して」という文節なのに対し，図 6 では「保護者として」となる． その変化によって，実際に学習・解析に使用する素性も，**表 6** のように機能表現を考慮したものに変化する．具体的には，係り元の文節が「して」から「保護者として」と変化することによって，係り元の主辞が「し」から「保護」に，係り元の語形が「て」から「として」に変化している．また，着目している係り元に係る文節も「保護者と」から「甥の」に変化している．このように学習・解析に使用する素性を機能表現を考慮したものにすることによって，機能表現を考慮した係り受け解析が実現できると考えられる．

5. 実験と考察

2.3 節の機能表現検出評価実験で対象とした 32 表現に対して，本論文で提案する係り受け解析器の学習および解析を行い，各ベースラインと性能比較をした．実験で使われた機能表現検出器は，2.3 節の実験で用いたデータセットで訓練を行ったものである．この際，素性は，形態素素性，チャンク素性，チャンク文脈素性を使用した．

表 5 機能表現の品詞体系

機能表現の分類		付与する品詞
接続詞型		接続詞-機能表現
助詞型	接続辞類	助詞-接続助詞-機能表現
	連用辞類	助詞-格助詞-機能表現
	連体辞類	助詞-連体化-機能表現
助動詞型		助動詞-機能表現

表 7 係り受け解析器用データセットの各統計量

	用法			全文数
	機能的用法	内容的用法	計	
京都テキストコーパス	4675	817	5492	38400
訓練用 101 文	164	319	483	101
評価用 1600 文	959	641	1600	1600

5.1 データセット

係り受け解析器の学習データとしては，京都テキストコーパス[14] を利用する．ここで，オリジナルの京都テキストコーパスには，機能表現の情報は付与されていないので，まず，京都テキストコーパス 38,400 文に存在する全ての機能表現に対して，用法ラベルを付与した．また，京都テキストコーパスでの生起頻度が低い機能表現に対しては，各機能表現について，各ラベルの用例が 5 例以上含まれるように，毎日新聞 (1995 年) から 101 文を別途収集し，文中の機能表現に対する用法ラベル，および，機能表現の係り元文節・係り先文節の情報を付与した．評価用データとしては，訓練データに使用していない 1600 文 (一つの表現につき 50 例文) を毎日新聞 (1995 年) から収集した．この評価用データでは，例文が対象としている 1 表現にのみ判定ラベルおよび係り受け関係が付与されており，それ以外の表現は評価対象とならないようになっている．これらの各データセットに含まれる各用法の数と，全文数を**表 7** に示す．

5.2 評価尺度

実験結果を評価する際の尺度には，以下の式で表される係り先精度，係り元精度を用いた．ただし，FE 文節とは，機能表現を含む文節を表している．

$$\text{係り先精度} = \frac{\text{係り先を正しく同定できた FE 文節数}}{\text{機能表現候補数}}$$

$$\text{係り元精度} = \frac{\text{係り元を正しく同定できた FE 文節数}}{\text{機能表現候補数}}$$

5.3 評価結果

本論文で提案している機能表現を考慮した係り受け解析器と各ベースラインの精度を**表 8** に示す．評価においては，(a) 京都テキストコーパスを訓練・評価データとする 10 分割交差検定，および，(b) 京都テキストコーパスおよび訓練用 101 文で訓練し，評価用 1600 文で評価，の二通りの評価を行った．表 8 中の「CaboCha(機能表現抜き)」は，ipadic 辞書に連語として登録されている機能表現の内，評価対象の機能表現にあたるものを機能表現を構成している形態素に分解し，学習し直している．「CaboCha(オリジナル)」は，他のモデルと同一の訓練データセットを用いて学習を行ったものである．また，

表 6 係り受けの学習・解析に使う素性の例

			機能表現を考慮しない	機能表現を考慮する
静的素性	係り元	主辞見出し	し	保護
		主辞品詞	動詞	名詞
		主辞品詞細分類	自立	サ変接続
		主辞活用	サ変・スル	*
		主辞活用形	連用形	*
		語形見出し	て	として
		語形品詞	助詞	助詞
		語形品詞細分類 1	接続助詞	格助詞
		語形品詞細分類 2	*	機能表現
		括弧の有無	0	0
		句読点の有無	0	0
		文節の位置 (文頭)	0	0
		文節の位置 (文末)	0	0
	係り先	主辞見出し	た	た
		主辞品詞	助動詞	助動詞
		主辞活用	特殊・タ	特殊・タ
		主辞活用形	基本形	基本形
		語形見出し	た	た
		語形品詞	助動詞	助動詞
		語形活用	特殊・タ	特殊・タ
		語形活用形	基本形	基本形
		括弧の有無	0	0
		句読点の有無	0	0
		文節の位置 (文頭)	0	0
		文節の位置 (文末)	0	0
	文節間	距離	2 以上 5 以下	2 以上 5 以下
		助詞	に	に
		括弧の有無	0	0
		句読点の有無	0	0
動的素性	着目している係り先に係る文節	語形見出し	に	に
	着目している係り元に係る文節	語形見出し	と	の
	着目している係り先が係る文節	主辞品詞	名詞	名詞

表 8 係り受け解析の評価結果 (%)

(a) 京都テキストコーパスを訓練・評価データとする 10 分割交差検定

		係り先精度	係り元精度
ベースライン	CaboCha(機能表現抜き)	90.7	80.0
	CaboCha(オリジナル)	91.1	81.8
提案手法	検出器出力使用	91.7	82.3
	正解用法ラベル使用	92.1	83.4

(b) 京都テキストコーパス+訓練用 101 文で訓練 評価用 1600 文で評価

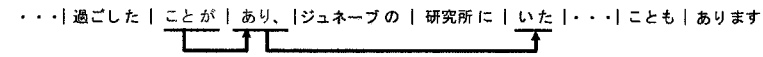
		係り先精度	係り元精度
ベースライン	CaboCha(機能表現抜き)	80.4	78.4
	CaboCha(オリジナル)	80.4	77.7
提案手法		80.9	78.5

機能表現を考慮した係り受け解析では，評価 (a) のみ，機能表現用法ラベルとして，2 章で述べた検出器により出力された結果を用いた場合，および，人手で付与した正解用法ラベルを用いた場合の二通りを評価した．

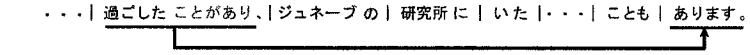
評価 (a) および (b) の最も大きな違いは，評価用データセットにおける各表現数の分布である．評価 (a) においては，京都テキストコーパスにおける機能表現数の分布がそのまま評価結果にも反映しており，例えば，機能表現候補としての出現頻度 1,000 以上の表現「として」「という」の影響が大きい．一方，評価 (b) においては，各機能表現候補の出現頻度は均等に 50 となっている．

評価 (a) においては，提案手法は，高頻度な表現において，ベースラインと同等かそれ以上の性能を達成しており，また，その他の表現に対する性能も含めて，総体として，ベースラインを上回る性能を達成している．特に，CaboCha(機能表現抜き) との比較においては，係り先精度，係り元精度とも，有意水準 8%で上回っている．また，正解用法ラベルを入力とした場合は，CaboCha(機能表現抜き) の係り先解析誤りのうちの 15%，および，係り元解析誤りのうちの 17%がそれぞれ改善できるのに対して，検出器の出力を入力とした場合は，それぞれ 11%および 12%の誤りが改善できる．

一方，評価 (b) においては，ベースラインに対する改善率はあまり大きくないが，表現ごとに係り受け解析性能を比較すると，提案手法の効果が大きい表現が存在することが分かる．具体的には，係り先精度がベースラインよりも 5 ポイント以上上回ったものは，「ところだ」，「ことがある」の 2 表現であり，5 ポイント以上下回ったものは存在しなかった．係り先精度が上昇した表現は，その表現を構成している形態素列を独立した形態素として扱うのではなく，一つの機能表現として検出していることが効果的に働いていると考えられる．例えば，「ことがある」を連用活用した「ことがあり」の場合，構成要素である形態素列を独立に扱うと．**図 7** (a) のように構成要素の一つである動詞「ある」の連用形「あり」が付近の

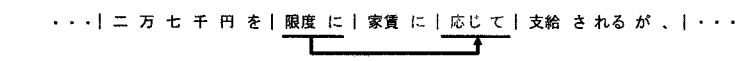


(a) ベースラインによる失敗例

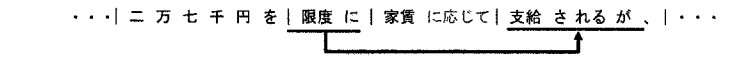


(b) 提案手法による成功例

図 7 係り先同定の改善例



(a) ベースラインによる失敗例



(b) 提案手法による成功例

図 8 係り元同定の改善例

動詞と並立に誤って係ってしまうことがある．それに対して，「ことがあり」を機能表現として扱った場合，図 7 (b) のように正しく係り先を推定できる．

一方，係り元精度がベースラインよりも 5 ポイント以上上回ったものは，「にせよ」，「に応じて」，「にかけ」，「をはじめ」の 4 表現であり，5 ポイント以上下回ったものは「にあたり」1 表現のみであった．係り元精度が上昇した表現も，係り先精度が上昇した表現同様，その表現を構成している形態素列を独立に扱うのではなく，一つの機能表現として検出していることが効果的に働いていると考えられる．例えば，「に応じて」の場合，構成要素である形態素列を独立に扱うと，**図 8** (a) のような例文において，「限度に」という文節が動詞を含む文節に係りやすいという特徴をもっているので，誤って「応じて」という文節に係ってしまう．それに対して，「に応じて」を機能表現として扱った場合，図 8 (b) のように，「限度に」の係り先を正しく推定することができる．係り元精度が下降した表現「にあたり」は，機能表現の検出が困難であり，機能表現検出の誤りが原因で係り元の同定精度が下がったと考えられる．

6. おわりに

本論文では，形態素を単位とするチャンク同定問題として機能表現検出タスクを定式化し，機械学習を利用して機能表現の検出を実現し，さらに，機能表現を考慮した係り受け解析を実現した．

参 考 文 献

1) John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, pp. 282–289, 2001.
2) E. Tjong Kim Sang. Noun phrase recognition by system combination. In *Proc. 1st NAACL*, pp. 50–55, 2000.
3) 土屋雅稔, 注連隆夫, 高木俊宏, 内元清貴, 松吉俊, 宇津呂武仁, 佐藤理史, 中川聖一. 機械学習を用いた日本語機能表現のチャンキング. 自然言語処理, Vol. 14, , 2007. (掲載予定).
4) 土屋雅稔, 宇津呂武仁, 松吉俊, 佐藤理史, 中川聖一. 日本語複合辞用例データベースの作成と分析. 情報処理学会論文誌, Vol. 47, No. 6, 2006.
5) 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 回年次大会発表論文集, pp. 584–587, 2005.
6) 国立国語研究所. 現代語複合辞用例集, 2001.
7) 高木俊宏, 注連隆夫, 土屋雅稔, 内元清貴, 松吉俊, 宇津呂武仁, 佐藤理史. 機械学習を用いた日本語機能表現のチャンキング. 言語処理学会第 12 回年次大会論文集, pp. 739–742, 2006.
8) 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.
9) 工藤拓, 山本薫, 松本裕治. Conditional random fields を用いた日本語形態素解析. 情報処理学会研究報告, 第 2004-NL-161 巻, pp. 89–96, 2004.
10) 工藤拓, 松本裕治. Support Vector Machine を用いた Chunk 同定. 自然言語処理, Vol. 9, No. 5, pp. 3–21, 10 2002.
11) 工藤拓, 松本裕治. チャンキングの段階適用による係り受け解析. 情報処理学会論文誌, Vol. 43, No. 6, pp. 1834–1842, 2002.
12) 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://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman/juman-5.1.tar.gz`.
13) 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://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp/knp-2.0.tar.gz`.
14) 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.
15) 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 ChaSen version 2.3.3 使用説明書. `http://chasen.aist-nara.ac.jp/chasen/doc/chasen-2.3.3-j.pdf`, 2003.