

日本語機能表現の自動検出と統計的係り受け解析への応用

注連 隆夫[†]・土屋 雅稔^{††}・松吉 俊^{†††,††††}・
宇津呂武仁^{††††}・佐藤 理史^{††††}

日本語には、「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機能表現という表現が存在する。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。そして、この表現が存在することによって、機能表現の検出は困難であり、機能表現を正しく検出できる機能表現検出器が必要とされている。そこで、本論文では、日本語機能表現を機械学習を用いて検出する手法を提案する。提案手法では、Support Vector Machine (SVM) を用いたチャンカー YamCha を利用して、形態素解析結果を入力とする機能表現検出器を構築する。具体的には、形態素解析によって得られる形態素の情報と、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、機能表現の前後の文脈の情報を学習・解析に使用することにより、F 値で約 93% という高精度の検出器を実現した。さらに、本論文では、機能表現検出器の解析結果を入力として、機能表現を考慮した係り受け解析器を提案する。提案手法では、Support Vector Machine (SVM) に基づく統計的係り受け解析手法を利用して、機能表現を考慮した係り受け解析器を構築する。具体的には、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、機能表現の情報を基に文節の区切りや係り先の情報を機能表現を考慮したものに変換した。そして、SVM に基づく統計的係り受け解析の学習・解析ツール CaboCha を用いて、変換したデータを学習し、機能表現を考慮した係り受け解析を実現した。評価実験では、従来の係り受け解析手法よりもよい性能を示すことができた。

キーワード：日本語機能表現、係り受け解析、機械学習

Automatic Detection of Japanese Compound Functional Expressions and its Application to Statistical Dependency Analysis

TAKAO SHIME^{†††}, MASATOSHI TSUCHIYA^{††}, SUGURU MATSUYOSHI^{†††,††††}, TAKEHITO UTSURO^{††††} and SATOSHI SATO^{††††}

The Japanese language has many compound functional expressions which consist

[†] 日本電気株式会社 C & C イノベーション研究所, NEC Corporation, C & C Innovation Research Laboratories

^{††} 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

^{†††} 京都大学大学院情報学研究科, Graduate School of Informatics, Kyoto University

^{††††} 名古屋大学大学院工学研究科, Graduate School of Engineering, Nagoya University

^{†††††} 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

of more than one words including both content words and functional words, e.g., “にあたって” and “をめぐって”. However, recognition and semantic interpretation of compound functional expressions are especially difficult because it often happens that one compound expressions may have both a literal content word usage and a non-literal functional usage. This paper proposes an approach of processing Japanese compound functional expressions by identifying them and analyzing their dependency relations through a machine learning technique. First, we formalize the task of identifying Japanese compound functional expressions in a text as a machine learning based chunking problem. Next, against the results of identifying compound functional expressions, we apply the method of dependency analysis based on the cascaded chunking model. In the experimental evaluation, we first show that the proposed method of chunking compound functional expressions significantly outperforms existing Japanese text processing tools. Next, we further show that, for many types of functional expressions, the cascaded chunking model applied to the results of identifying compound functional expressions outperforms the one applied to the results without identifying compound functional expressions.

Key Words: *Japanese Compound Fuctional Expression, Dependency Analysis, Machine Learning*

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.

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. このような表現においては, 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を識別する必要がある(土屋, 宇津呂, 松吉, 佐藤, 中川 2006). 以下, 本論文では, 文(1), (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.

機能表現検出は, 日本語解析技術の中でも基盤的な技術であり, 高カバレッジかつ高精度な技術を確立することにより, 後段の様々な解析や応用の効果が期待できる.

一例として, 以下の例文を題材に, 機能表現検出の後段の応用として機械翻訳を想定した場合を考える.

(3) 私は、彼の車について走った。

(4) 私は、自分の夢について話した。

文(3)では、下線部は内容的用法として働いており、文(4)では、下線部は機能的用法として働いており、それぞれ英語に訳すと、文(5)、文(6)となる。

(5) I drove following his car.

(6) I talked about my dream.

下線部に注目すれば分かる通り、英語に訳した場合、内容的用法と機能的用法で対応する英単語が異なっている。このように内容的用法と機能的用法で対応する英単語が異なるので、機能表現検出のタスクは、機械翻訳の精度向上に効果があると考えられる。

また、機能表現検出の後段の解析として格解析を想定する。格解析は、用言とそれがとる格要素の関係を記述した格フレームを利用して行われる。

(7) 私は、彼の仕事について話す。

「について」という機能表現を含む文(7)において、格解析を行う場合、機能表現を考慮しなければ、「仕事」と「話す」の関係を検出することができず、「私は」と「話す」の関係がガ格であることしか、検出できない。それに対して、「について」という機能表現を考慮することができれば、「仕事」と「話す」の機能的な関係を「について」という機能表現が表現していることが検出することができる。このことから、機能表現検出の結果は、格解析の精度向上に効果があると考えられる。

さらに、以下の例文を題材にして、機能表現検出の後段の解析として係り受け解析を想定する。

(8) 2万円を 限度に 家賃に 応じて 支給される。

(9) 2万円を 限度に 家賃に 応じて 支給される。

文(8)、文(9)における空白の区切りは、それぞれ、機能表現を考慮していない場合の文節区切り、機能表現を考慮した場合の文節区切りを表している。この例文において、「限度に」という文節の係り先を推定する時、「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ため、文(8)の場合、「応じて」という文節に係ってしまう。それに対して、文(9)では、「に 応じて」を機能表現として扱っているため、「限度に」の係り先を正しく推定できる。このようなことから、機能表現のタスクは、格解析の精度向上に効果があると考えられる。本論文では、これら3つの応用研究の内、係り受け解析への機能表現検出の適用方法を考えた。

日本語の機能表現として認定すべき表記の一覧については、いくつかの先行研究が存在する。(森田 松木 1989)は、450種類の表現を、意味的に52種類に分類し、機能的に7種類に分類している。(松吉, 佐藤, 宇津呂 2006)は、森田らが分類した表現の内、格助詞、接続助詞および助動詞に相当する表現について、階層的かつ網羅的な整理を行い、390種類の意味的・機能的に異なる表現が存在し、その異形は13690種類に上ると報告している。(土屋他 2006)は、森田らが分

類した表現の内、特に一般性が高いと判断される 337 種類の表現について、新聞記事から機能表現候補を含む用例を無作為に収集し、人手によって用法を判定したデータベースを作成している。このデータベースによると、機能表現候補が新聞記事（1 年間）に 50 回以上出現し、かつ、機能的な意味で用いられている場合と、それ以外の意味で用いられている場合の両方が適度な割合で出現する表現は、59 種類である。本論文では、この 59 種類の表現を当面の検討対象とする。

まず、既存の解析系について、この 59 種類の表現に対する取り扱い状況を調査したところ、59 種類の表現全てに対して十分な取り扱いがされているわけではないことが分かった¹。59 種類の表現の内、形態素解析器 JUMAN(黒橋 河原 2005a) と構文解析器 KNP(黒橋 河原 2005b) 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 24 種類である。また、形態素解析器 ChaSen(松本, 北内, 山下, 平野, 松田, 高岡, 浅原 2003) と構文解析器 CaboCha(工藤 松本 2002b) の組み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 20 種類である。

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。

まず、検出器の実現方法を考えた場合、検出対象である機能表現を形態素解析用辞書に登録し、形態素解析と同時に機能表現を検出する方法と、形態素解析結果を利用して機能表現を検出する方法が考えられる。現在、広く用いられている形態素解析器は、機械学習的なアプローチで接続制約や接続コストを推定した辞書に基づいて動作する。そのため、形態素解析と同時に機能表現を検出するには、既存の形態素に加えて各機能表現の接続制約や接続コストを推定するための、機能表現がラベル付けされた大規模なコーパスが必要になる。しかし、検出対象の機能表現が多数になる場合は、作成コストの点から見て、そのような条件を満たす大規模コーパスを準備することは容易ではない。

形態素解析と機能表現検出が独立に実行可能であると仮定し、形態素解析結果を利用して機能表現を検出することになると、前述のような問題を避けられる。そこで、機能表現の構成要素である可能性がある形態素が、機能表現の一部として現れる場合と、機能表現とは関係なく現れる場合で、接続制約が変化しないという仮定を置いた上で、人手で作成した検出規則を形態素解析結果に対して適用することにより機能表現を検出する手法が提案されてきた(?)。しかし、これらの手法では、検出規則を人手で作成するのに多大なコストが必要となり、検出対象とする機能表現集合の規模の拡大に対して追従が困難である。

そこで、本論文では、機能表現検出と形態素解析は独立に実行可能であると仮定した上で、機能表現検出を形態素を単位とするチャンク同定問題として定式化し、形態素解析結果から機械

¹ 詳しくは、3.2.3 節を参照。

学習によって機能表現を検出するアプローチ (土屋, 注連, 高木, 内元, 松吉, 宇津呂, 佐藤, 中川 2007) をとる. 機械学習手法としては, 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によって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な Support Vector Machine (SVM)(Vapnik 1998) を用いる. 具体的には, SVM を用いたチャンカー YamCha(工藤 松本 2002a) 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. ただし, 形態素解析用辞書に「助詞・格助詞・連語」や「接続詞」として登録されている複合語が, 形態素解析結果中に含まれていた場合は, その複合語を, 構成要素である形態素の列に置き換えた形態素列を入力とする. また, 訓練データとしては, 先に述べた 59 表現について人手で用法を判定したデータを用いる. 更に, このようにして実装した機能表現検出器は, 既存の解析系および (土屋, 宇津呂, 佐藤, 中川 2005) が提案した人手で作成した規則に基づく手法と比べて, 機能表現を高精度に検出できることを示す.

次に, 機能表現を考慮した係り受け解析器の実現方法としては, 既存の解析系である KNP と CaboCha を利用する方法が考えられる. KNP を利用する場合は, 新たに機能表現を考慮した係り受け規則を作成する必要がある. それに対して, CaboCha を利用する場合は, 現在使用されている訓練用データ (京都テキストコーパス (黒橋 長尾 1997)) を機能表現を考慮したものに自動的に変換すればよい. そこで, 本論文では, CaboCha の学習を機能表現を考慮した訓練データで行うことによって, 機能表現を考慮した係り受け解析器を実現する. 訓練データの作成には, 訓練の対象となる文の係り受け情報と文に存在する機能表現の情報を利用する.

本論文の構成は以下の通りである. 2 節で, 本論文の対象とする機能表現と, その機能表現候補の用法を表現するための判定ラベルについて述べる. 3 節で, 機能表現検出をチャンク同定問題として定式化し, SVM を利用した機能表現のチャンキングについて説明し, 機能表現検出器の検出性能の評価を行い, この検出器が, 既存の解析系および人手によって規則を作成した手法と比べ, 機能表現を高精度に検出できることを示す. 4 節では, 機能表現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について説明を行い, 機能表現を考慮した係り受け解析器と従来の係り受け解析器を使った機能表現を考慮した最適な係り受け解析について述べ, 実際に機能表現を考慮した係り受け解析の評価を行う. 5 節では, 関連研究について述べ, 最後に 6 節で結論を述べる.

2 機能表現およびその用法

2.1 用例データベース

森田ら (森田, 松木 1989) は, 機能表現の中でも特に「単なる語の接続ではなく, 表現形式全体として, 個々の構成要素のプラス以上の独自の意味が生じている」表現を**複合辞**と呼び, 個々の構成要素の意味から構成的に表現形式全体の意味を説明できるような表現とは区別している.

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	例文
B	不適切				(10) … と谷川王将は気にかけている.
Y	適切	不一致			(11) 地球上では、人口の増加、異常気象が心配されている.
C	適切	一致	内容的	内容的用法	(12) まな板にとっていていねいに… みそ汁の実にするのである.
F	適切	一致	機能的	複合辞用例集の用法	(13) 受験などでは倍率が上がったところで入学金があがることはない.
A	適切	一致	機能的	接続詞的用法	(14) ところで、全国の桜の名所では近年、樹勢の衰えが目立ち、…
M	適切	一致	機能的	他の機能的用法	(15) 浜ノ島はあと一步のところで勝ち星に結び付かず負け越した.

現代語複合辞用例集 (国立国語研究所 2001) (以下、**複合辞用例集**と呼ぶ) は、主要な 125 種類の複合辞について、用例を集成し、説明を加えたものである。

日本語複合辞用例データベース (土屋他 2006) (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。用例データベースは、複合辞用例集に収録されている 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 個の用例を収録している。また、用例は、毎日新聞 1995 年から収集されている。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を、表 1 および次節に示す 6 種類の判定ラベルのうちから人手で判定し、付与している。

2.2 判定ラベル体系

判定ラベルとは、機能表現候補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り、用例データベースでは表 1 の通り、6 種類のラベルが設定されている。以下、個々の判定ラベルについて説明する。

用例データベースでは、IPA 品詞体系 (THiMCO97) の形態素解析用辞書 (浅原 松本 2003) に登録されている語から、「助詞・格助詞・連語」として登録されている語を取り除いた残りの語を、語としている。そして、ある機能表現候補が、1 個以上の語、複合辞または慣用表現からなる列である場合、その候補は判定単位として適切であるが、それ以外の場合は、その候補は判定単位として不適切であるとして、判定ラベル B を付与している。例えば、表 1 中の文 (10) に含まれる機能表現候補「にかけて」は、「心配する」という意味の慣用表現「気にかける」の一部が活用した形であり、先に述べた条件を満たしていない。したがって、文 (10) には、判

定ラベル B が付与される。

判定ラベル Y は, 機能表現候補の読みが, 判定対象となっている機能表現の読みと一致していないことを表す。例えば, 「A うえで B」という形で, 「A した後で B」という出来事の継起関係を表す機能表現「うえで」の用例として表 1 中の文 (11) を判定する場合を考える。この場合, 機能表現候補の読み「じょうで」と, 判定対象となっている機能表現の読み「うえで」が一致していないので, 判定ラベル Y を付与する。

判定ラベル C は, 機能表現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, 表 1 中の文 (12) の機能表現候補に含まれる動詞「とる」は本来の意味で内容的に働いているので, 判定ラベルとして C を付与する。

判定ラベル F, A, M は, 機能表現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, 機能表現候補が複合辞用例集で説明されている用法で働いていることを表し, 判定ラベル A は, 機能表現候補が接続詞的に働いていることを表す。判定ラベル M は, これら以外の機能的な働きをしていることを表す。例として, 機能表現候補「ところで」の用例として表 1 中の文 (13) ~ (15) を判定する場合を考える。文 (13) では, 複合辞用例集で説明されている通りに逆接の働きをしているので, 判定ラベル F を付与する。文 (14) では, 文頭で接続詞的に働いているので, 判定ラベル A を付与する。文 (15) では, 形式名詞「ところ」を含めて機能的に働いているので, 判定ラベル M を付与する。本論文では, 判定ラベル F, A, M が付与される機能表現候補を検出対象とする。

3 機能表現検出

3.1 SVM を用いたチャンキングによる機能表現検出

3.1.1 Support Vector Machines

サポートベクトルマシンは, 素性空間を超平面で分割することによりデータを 2 つのクラスに分類する二値分類器である (Cristianini and Shawe-Taylor 2000; Kudoh 2000)。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。すなわち, 以下の識別関数 $f(x)$ の値によってクラスを判別することと等価である。

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

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

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

ここで $\mathbf{x}$ は識別したい事例の文脈 (素性の集合), $\mathbf{x}_i$ と $y_i (i = 1, \dots, l, y_i \in \{1, -1\})$ は学習データの文脈とクラスである. また, 関数 $\text{sgn}(x)$ 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る. 各 α_i は, 式 (3) と式 (4) の制約のもとで式 (2) の $L(\boldsymbol{\alpha})$ を最大にするものである.

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

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

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

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, 本論文では次式で定義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.

予備実験を行い, 次数 d の値として 1, 2, 3 の 3 通りを検討した. $d = 2, 3$ とした場合は F 値に大きな差はなかったが, $d = 1$ とすると F 値がかなり悪化した². ただし, $d = 3$ とした場合は, $d = 2$ とした場合に比べて, 学習時間がかなり増加したため, 本論文では, 次数 d の値として 2 を用いる. また, 予備実験において, マージン C の値として 1, 0.1, 0.01, 0.001, 0.0001 の 5 通りを検討したところ, F 値に大きな差が見られなかったため, 本論文ではマージン C の値として 1 を用いる.

3.1.2 チャンクタグの表現法

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するという手順で, 機能表現検出を行う. 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.

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 (Tjong Kim Sang 2000) 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

² 評価尺度 (F 値) については 3.2.2 節を参照.

表 2 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-機能	B-自立
	チャンクの先頭以外	I-機能	I-自立
それ以外の形態素		O	

ただし, 本論文では IOB2 フォーマットを, さらに表 2 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. この表において, 機能的用法とは, 用例データベースで設定されている判定ラベルのうち, ラベル F, A, M のいずれかが付与されたものを表し, 内容的用法とは, 判定ラベルのうち, ラベル C, Y, B のいずれかが付与されたものを表している. 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.

SVM は二値分類器であるため, そのままでは, 2 クラスの分類しか扱えない. 本論文のようにクラス数が 3 以上の場合には, 複数の二値分類器を組み合わせて拡張する必要がある. 本論文では, 拡張手法としては, 広く利用されているペアワイズ法を用いる. ペアワイズ法とは, N 個のクラスに属するデータを分類する時, 異なる 2 つのクラスのあらゆる組み合わせに対する二値分類器を作り, 得られた $N(N-1)/2$ 個の二値分類器の多数決により, クラスを決定する方法である.

3.1.3 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.

$$F_i = \langle MF(m_i), CF(i), OF(i) \rangle \quad (6)$$

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 (浅原, 松本 2003) 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 $m_j \dots m_i \dots m_k$ からなる機能表現候補 E が存在したとする.

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

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.

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

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

機能表現検出においては, 1 つの文中に, 複数の機能表現候補が部分的に重複して現れる場合を考慮する必要がある. ここでは, そのような場合のチャンク素性とチャンク文脈素性の付与方法について考える. 複数の機能表現候補が部分的に重複して現れている場合, 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, 後者の方法を採用, 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 (16) には, 「なくてははいけません」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.

(16) 慎重になくてははいけません.

この場合, 「なくてははいけません」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 (17) 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.

(17) 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 $F_{i-2}, F_{i-1}, F_i, F_{i+1}, F_{i+2}$

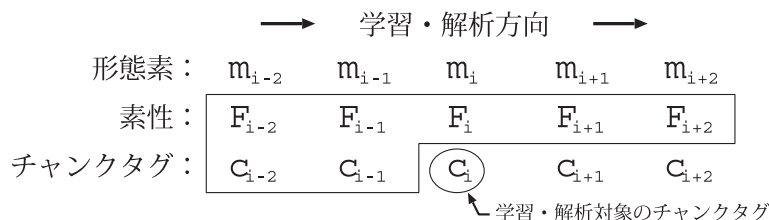


図 1 YamCha の学習・解析

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.

3.2 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.

3.2.1 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判定ラベルが付与されたデータが必要である. 本論文では, 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 59 表現を対象とする. そして, これらの 59 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 2583 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判定ラベルを付与した. さらに, この例文の内, 京都テキストコーパスに含まれる文と重複する 154 文を除いた. 本論文では, この 2429 文 (各表現について 20 用例以上収録) を機能表現検出器の訓練データとして使用する.

ただし, 用例データベースでは, 機能表現候補の先頭と末尾が形態素境界と一致しない候補にも判定ラベルが付与されているが, 本論文では, 形態素解析結果に基づいて機能表現を検出する立場をとるため, そのような機能表現候補に対する判定ラベルは取り除くことにする. 具体的には, 以下のような処理を行った. 最初に, 用例データベースに収録されている用例を, IPA 品詞体系の形態素解析用辞書に基づいて動作する形態素解析器 ChaSen を用いて形態素解析した. 次に, 形態素解析結果中に, 形態素解析用辞書に「助詞・格助詞・連語」や「接続詞」として登録されており, かつ実験の対象である 59 表現となる複合語が含まれていた場合は, その複合語を, 構成要素である形態素の列に置き換えた. このようにして得られた形態素解析結果と

表 3 データセットの各統計量

	用法		計	全形態素数
	機能的用法	内容的用法		
訓練データ（毎日新聞 1995 年）	1918	1165	3083	85806
評価データ（京都テキストコーパス）	5744	1959	7703	239512

機能表現候補を照合し、先頭と末尾が形態素境界と一致しなかった判定ラベルを取り除いた。

また、機能表現検出器の評価データとしては、京都テキストコーパスに収録されている文を対象とし、その文に含まれている全ての機能表現候補に対して、判定ラベルを付与したものを使用した。

訓練・評価データに含まれる各用法の数と、全形態素数を表 3 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。また、評価データ（京都テキストコーパス）における機能表現候補の分布は、表 4 の通りである。

表 4 には、京都テキストコーパスにおける機能表現の分布以外に、機能表現の用例データベースにおける分類、その分類に基づいた係り受け解析の学習の際に使用する品詞体系の情報が示されている。機能表現の分類には、接続詞相当の働きをするもの（接続詞型）、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）の 3 種類存在する。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。係り受け解析の学習の際に使用する品詞体系は、上で述べた機能表現の分類に基づいて作成されている。

また、表 4 には、「といっても」、「とはいえ」など、接続詞型と助詞型の二つの分類に重複して登場している表現がある。これは、「といっても」などの機能表現候補は、接続詞型、助詞型のどちらの機能表現にもなりうるからである。

3.2.2 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。

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

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

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

注連, 土屋, 松吉, 宇津呂, 佐藤

日本語機能表現の自動検出と統計的係り受け解析への応用

表 4 京都テキストコーパスにおける機能表現候補の出現頻度

機能表現の分類		品詞	表現	用例数	
				表現単位	品詞単位
助詞型	接続辞類	助詞—接続助詞—機能表現	といって といっても とはいえ とすると というど といえは となると といいながら かと思うと と思えは と思ったら うえに うえは あとで あとに くせに ところを ところが おりから かわりに ものを ものなら にしても にしろ にせよ	89 56 82 10 18 47 20 9 1 6 4 67 17 12 174 1 21 28 6 20 99 6 38 16 20	867
	連用辞類	助詞—格助詞—機能表現	にあたって にあたり にあって に限り に限って に応じて に応じ にかけて にいたるまで に従って に従い につき につけて につけ にとり にかけ として をはじめ をとおして からして	48 22 26 11 11 36 19 60 8 14 14 9 5 14 5 25 1672 105 32 8	2144
	連体辞類	助詞—連体化—機能表現	に応じた うえでの をめぐる という といった	21 11 205 3346 152	3735
接続詞型		接続詞—機能表現	といって といっても とはいえ とすると と思ったら ところが かわりに	2 4 20 2 1 139 1	169
助動詞型		助詞—機能表現	ところだ ことがある ことがない ことはない に決っている ていい て仕方がない 得る なくてはならない	46 73 23 73 1 171 6 444 50	887

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

3.2.3 既存の解析系に対する評価基準

既存の解析系 (JUMAN/KNP および ChaSen/CaboCha) は、形態素解析および構文解析段階で処理が必要となる機能表現を、部分的に処理の対象としている。しかし、明示的に機能表現を取り扱うという立場は取っていないため、機能表現のチャンキングというタスクに対する既存の解析系の性能を評価するには、その出力をどのように解釈するかを定めておく必要がある。

形態素解析器 JUMAN と構文解析器 KNP の組み合わせでは、機能表現は以下のように処理される。最初に、接続詞として形態素解析用辞書に登録されている機能表現は、形態素解析時に検出される。次に、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、機能表現である可能性を考慮した解析を行う。一方、IPA 品詞体系 (THiMCO97) の形態素解析用辞書 (浅原, 松本 2003) を用いた形態素解析器 ChaSen と、京都テキストコーパス (黒橋, 長尾 1997) から機械学習したモデルを用いた構文解析器 CaboCha の組み合わせでは、機能表現は以下のように処理される。最初に、形態素解析用辞書に「助詞・格助詞・連語」や「接続詞」として登録されている機能表現は、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、機能的な表現として解析される。

既存の解析系でも、一部の機能表現については、機能的な働きをしていることを考慮した解析が行われているが、その対応状況は不十分である。判定ラベル F, A, M のいずれかが付与されている用例の内、少なくとも 1 つの用例が、機能的に働いている可能性を考慮して解析され、かつ、判定ラベル C, Y, B のいずれかが付与された用例の内、少なくとも 1 つの用例が、機能的に働いている可能性を考慮せずに解析されている場合、その機能表現は、用法が正しく区別される可能性があるとする。用例データベースに 50 用例が収録されている表現で、かつ、機能的な意味で用いられている場合と、それ以外の意味で用いられている場合の両方が適度な割合で出現する表現は、59 種類ある³。その内、JUMAN/KNP によって用法が正しく区別される可能性がある表現は、23 種類である。一方、ChaSen/CaboCha によって用法が正しく区別される可能性がある表現は 21 種類である。また、用例データベースに収録されている 337 表現全体では、新聞上の実際の用法の割合に関係なく識別が必要と思われる表現は、111 種類である。その内、JUMAN/KNP によって用法が正しく区別される可能性がある表現は 43 種類、ChaSen/CaboCha によって用法が正しく区別される可能性がある表現は 40 種類である。

³ ここでの機能表現の種類数には、(土屋他 2007) における記述とは差異があるが、これは、本論文においては、「機能的な意味で用いられている場合と、それ以外の意味で用いられている場合の両方が適度な割合で出現する」という条件の認定方法の改訂を行ったためである。

3.2.4 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5に示す。表5において、「頻度最大の判定ラベル」とは、全ての候補部分に対して頻度最大の判定ラベル（機能的用法）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、(土屋他 2005) による検出性能である。

表5中の「CRF を用いた検出器」は、Conditional Random Fields (CRF)(Lafferty, McCallum, and Pereira 2001) 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。本論文では、CRF による学習・解析用ツールとして CRF++⁴ 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 (Kudo, Yamamoto, and Matsumoto 2004)。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5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。

表5を見ると、「JUMAN/KNP」、「ChaSen/CaboCha」が他の手法に比べて著しく性能が悪いのがわかる。これは、3.2.3 節で述べたように、「JUMAN/KNP」、「ChaSen/CaboCha」が取り扱っている機能表現が、本実験の対象である 59 表現の内、23 表現、21 表現となっているのが、一つの原因である。もう一つの原因は、評価対象の大部分を占める「という」という表現

表 5 各検出器の検出性能 (%)

		精度	再現率	F 値	判別率
ベース ライン	頻度最大の判定ラベル	74.6	100	85.5	74.6
	JUMAN/KNP	85.8	40.5	55.0	58.4
	ChaSen/CaboCha	85.2	26.7	40.6	51.1
人手作成の規則による検出器		83.7	84.1	83.9	79.9
CRF を用いた検出器		90.9	92.2	91.6	87.3
SVM を 用いた 検出器	形態素素性	90.4	91.4	90.9	86.7
	形態素素性, チャンク素性	91.8	93.0	92.4	88.9
	形態素素性, チャンク素性, チャンク文脈素性	91.4	94.6	92.9	89.3

⁴ <http://chasen.org/~taku/software/CRF++/>

に対する再現率が、両解析系において、著しく低いということである。

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、評価用データにおいて10用例以上存在し、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(18)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。それに対して、文(19)および文(20)に含まれる「にあたり」は、内容的に用いられている。

(18) 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。

(19) お神酒の瓶が女性にあたり、けがをする事故があった。

(20) 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を用いた検出器は、文(18)と文(19)の用法を内容的用法として、また、文(20)の用法を機能的用法として検出してしまい、用法を正しく判定できたのは文(19)のみであった。仮に、文(18)と文(19)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(20)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5に示したように適切に検出することができた。

4 機能表現を考慮した係り受け解析器

4.1 SVMを用いた統計的係り受け解析

本論文では、SVMを用いた統計的係り受け解析手法(工藤, 松本 2002b)を利用して係り受け解析を行っている。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に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、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 $b_1, b_2, \dots, b_m$ を表しており、係り受けパターン列 D とは、 $Dep(1), Dep(2), \dots, Dep(m-1)$ 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によって、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、係り受けが未定であることを示す 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否かを判定。係

る場合は D タグを付与. 文末から 2 番目の文節には無条件に 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.

図 2 では, 入力として「彼は彼女の温かい真心に感動した。」という文を文節単位に区切ったものが与えられている. そして, それぞれの文節に対して, 係り受けが未定であることを示す O タグが付与される. その後, O タグが付与されている文節に対し, 直後の文節に係るか否かを判定する (文末から 2 番目の文節は無条件に D タグを付与). すると, 「温かい」, 「真心に」という文節が直後の文節に係ると推定されるので, D タグが付与される. その後, 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ので, 「温かい」という文節を削除する. こ

初期化

Input:	彼は	彼女の	温かい	真心に	感動した。
Tag:	0	0	0	0	0

Input:	彼は	彼女の	温かい	真心に	感動した。
Tag:	0	0	D _{削除}	D	0

Input:	彼は	彼女の	真心に	感動した。	
Tag:	0	D _{削除}	D	0	

Input:	彼は	真心に	感動した。		
Tag:	0	D _{削除}	0		

Input:	彼は	感動した。			
Tag:	D _{削除}	0			

Input:	感動した。				
Tag:	0				終了

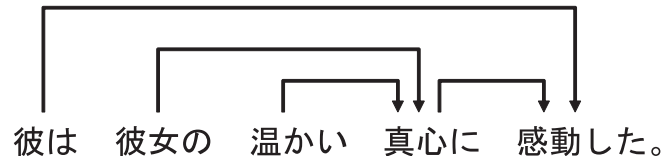


図 2 係り受け解析の流れ

の文節を削除できる理由としては、削除される文節は、非交差条件を考慮すると、他の文節から修飾されることはなく、それ自身の係り先もすでに同定されているため、係り受け候補として考慮する必要がなくなるためである。以上の作業を、入力「感動した。」という文節のみになるまで続けると、「彼は」が「感動した。」に、「彼女の」が「真心に」に、「温かい」が「真心に」に、「真心に」が「感動した。」に係ると判定することができる。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している。つまり、学習で使われる文節のセットは、上のアルゴリズムにおいて隣り合う文節のみであるので、負例が不必要に増えるのを防ぐことができる。

SVMの学習・解析に使用する素性は、表6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、文末に一番近い形態素のことを指す。

具体的に図3の文において、「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・解析に使われる素性について見てみる。まず、係り元、係り先の文節である「して」と「参加した」の主辞、語形の情報と、各文節における括弧の有無、句読点の有無、文節の位置（文頭、文末）が素性として使用される。次に文節間の素性として、文節の距離、文節の間に存在する全ての助詞の見出し、文節間の括弧の有無、文節間の句読点の有無が使用される。「して」と「参加した」の間には、「運動会に」という文節が存在している。よって、文節の距離として

表 6 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

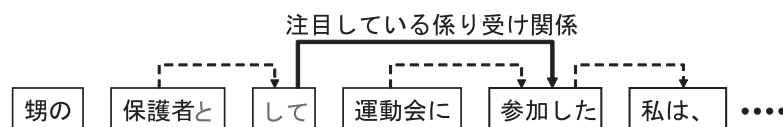


図 3 係り受け解析例

注連, 土屋, 松吉, 宇津呂, 佐藤

日本語機能表現の自動検出と統計的係り受け解析への応用

は, 「2 以上 5 以下」(素性として 1, 2 以上 5 以下, 6 以上の 3 通りの素性を選択) が使用される. 文節の間に存在する全ての助詞の見出しとしては, 「運動会に」に含まれる「に」が使用される. 括弧の有無は, 「運動会に」には括弧が含まれていないので「0」, 句読点の有無も, 句読点が含まれていないので「0」が使用される. 動的素性としては, 係り先文節「参加した」に係る文節「運動会に」の語形見出し「に」と, 係り元文節「して」に係る文節「保護者と」の語形見出し「と」と, 係り先文節「参加した」に係る文節「私は,」の主辞品詞「名詞」が使用される. 以上の素性の一覧を表 7 に示す.

表 7 係り受けの学習・解析に使う素性の例

静的素性	係り元	主辞見出し	し
		主辞品詞	動詞
		主辞品詞細分類	自立
		主辞活用	サ変・スル
		主辞活用形	連用形
		語形見出し	て
		語形品詞	助詞
		語形品詞細分類	接続助詞
		括弧の有無	0
		句読点の有無	0
		文節の位置 (文頭)	0
		文節の位置 (文末)	0
	係り先	主辞見出し	た
		主辞品詞	助動詞
		主辞活用	特殊・タ
		主辞活用形	基本形
		語形見出し	た
		語形品詞	助動詞
		主辞活用	特殊・タ
		主辞活用形	基本形
		括弧の有無	0
		句読点の有無	0
		文節の位置 (文頭)	0
		文節の位置 (文末)	0
	文節間	距離	2 以上 5 以下
		助詞	に
		括弧の有無	0
		句読点の有無	0
動的素性	着目している係り先に係る文節	語形の見出し	に
	着目している係り元に係る文節	語形の見出し	と
	着目している係り先が係る文節	主辞品詞	名詞

4.2 機能表現を考慮した係り受け解析

次に、本論文で提案する、機能表現を考慮した係り受け解析の流れを図4に示す。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。

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4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（国立国語研究所 2001）に掲載されている各機能表現と品詞分類との対応に基づいて、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。

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。その際に、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している。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、既存の係り受け情報付と済みコーパスから得られる係り受け関係の情報である。もう一つは、対象文における機能表現の情報である。この二つの情報を用いて図5の流れで、訓練データを作成し、学習を行っている。

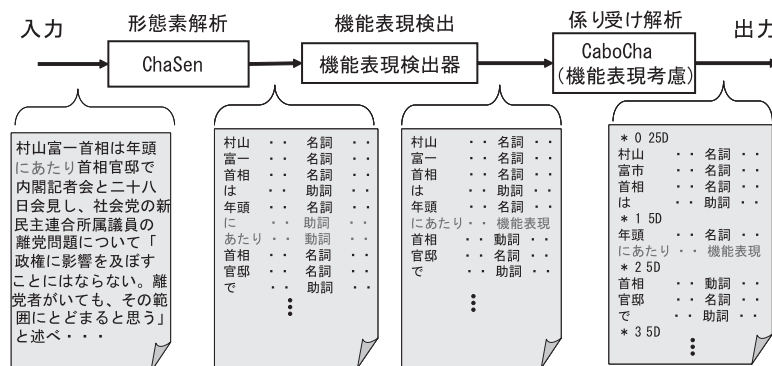


図4 機能表現を考慮した係り受け解析

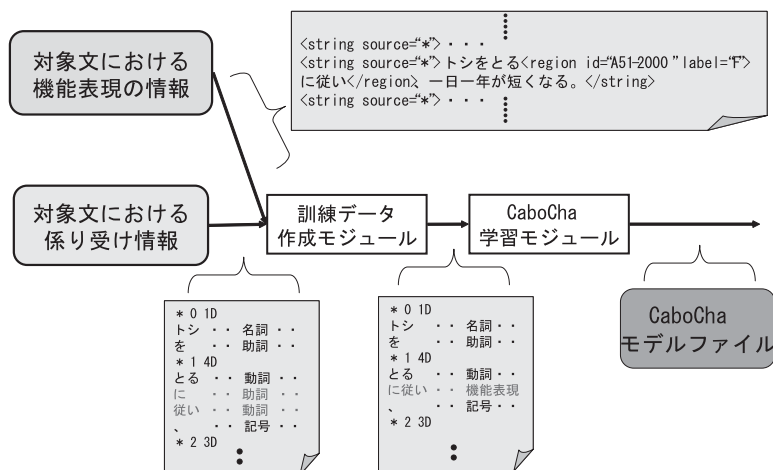


図5 機能表現を考慮した係り受け解析器の学習の流れ

図5の訓練データ作成モジュールでは、末尾の文節から順に以下の手順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。
- 3a. 助詞・助動詞型の機能表現の場合で、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. 接続詞型の機能表現の場合で、一文節が機能表現のみで構成されない場合は、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。

図6に、機能表現を考慮した係り受け解析の訓練データ作成の例を示す。図6中には、「にあたり」という機能表現が存在している。よって、まず格助詞「に」と動詞「あたる」の連用形「あたり」の連結を行う。それに伴い、「年頭に」という文節と「あたり」という文節の連結を行う。連結された「年頭にあたり」という文節の係り先は、「あたり」の係り先を採用する。次に、「年頭にあたり」以降の文節の文節IDと、「年頭にあたり」以降の文節に係る文節の係り先文節IDに対して変更を加える。このような作業をすることによって、機能表現を考慮していない係り受け解析の訓練データを、機能表現を考慮したものに変換していく。

機能表現を考慮しない係り受け解析の学習(図7)と機能表現を考慮した係り受け解析の学習(図8)の間では、学習に使用する素性が異なる。以下では、図7における「して」という文節、

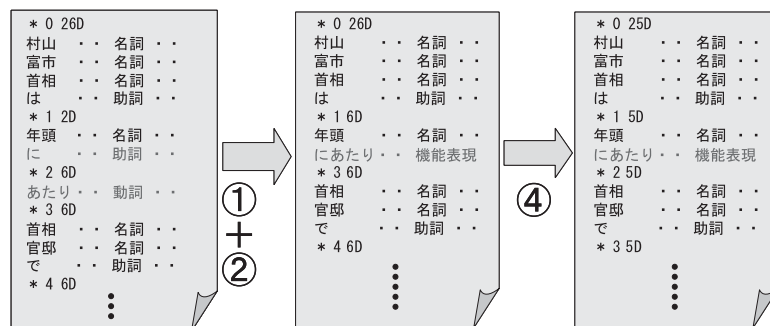


図 6 機能表現を考慮した係り受け解析の訓練データ作成の例

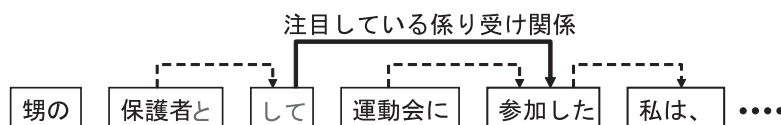


図 7 係り受け解析例（機能表現考慮せず）

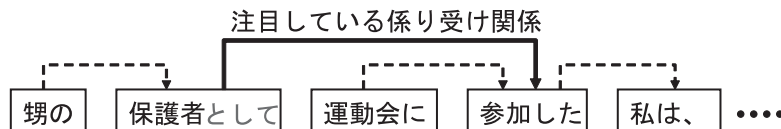


図 8 係り受け解析例（機能表現考慮）

および、図 8 における「保護者として」という文節と、「参加した」という文節の間の係り受け関係に注目する。まず、図 8 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。それによって、注目する係り受け関係の係り元文節が、図 7 では「して」という文節なのに対し、図 8 では「保護者として」となる。この違いによって、表 8 に示すように、実際に学習・解析に使用する素性の間にも差違が生じる。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、係り元の文節に係る文節も「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される。

4.3 実験と考察

本論文で提案する係り受け解析器に対して、学習および解析を行い、各ベースラインと性能比較をした。この際、対象とする表現は、機能表現検出器が対象としていた 59 表現である。実験で使われた機能表現検出器は、3.2 節の実験の訓練データで訓練を行ったものである。この

注連, 土屋, 松吉, 宇津呂, 佐藤

日本語機能表現の自動検出と統計的係り受け解析への応用

表 8 係り受けの学習・解析に使う素性の変化

			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

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.

4.3.1 データセット

係り受け解析器の訓練データとしては, 京都テキストコーパス (黒橋, 長尾 1997) 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

表 9 係り受け解析器用データセットの各統計量

用法		計	全文数
機能的用法	内容的用法		
5744	1959	7703	38400

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、判定ラベルを付与した。これらのデータセットに含まれる各用法の数と、全文数を表 9 に示す。

4.3.2 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。

$$\text{係り先精度} = \frac{\text{係り先を正しく同定できた文節数}}{\text{機能表現候補を含む文節数}}$$

$$\text{係り元精度} = \frac{\text{係り元を正しく同定できた文節数}}{\text{機能表現候補を含む文節数}}$$

4.3.3 評価結果および考察

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 10 に示す。評価においては、京都テキストコーパスを訓練・評価データとする 10 分割交差検定を行った。表 10 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、CaboCha の訓練を再度行ったものである。それらの機能表現は、59 表現中「ところが」、「にあたって」、「にあたり」、「にかけて」、「に従い」、「につき」、「につけ」、「にとり」、「にかけ」、「として」、「をめぐる」、「という」、「といった」の 13 表現である。「CaboCha (オリジナル)」は、上記の連語に対して構成形態素への分解を行わず、CaboCha の訓練を再度行ったものである。また、機能表現を考慮した係り受け解析では、機能表現判定ラベルとして、3 節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判定ラベルを用いた場合の二通りを評価した。

表 10 を見ると、提案手法は、係り先精度については、ベースラインとの差を見ることができなかったが、係り元精度については、ベースラインと比べ統計的に有意な改善 (有意水準 5%)

表 10 係り受け解析の評価結果 (%)

		係り先精度	係り元精度
ベースライン	CaboCha (機能表現抜き)	88.0	72.5
	CaboCha (オリジナル)	87.6	73.9
提案手法	検出器出力使用	88.0	74.0
	正解判定ラベル使用	88.1	74.4

が見られた⁵。よって、機能表現検出や、機能表現を考慮することが、係り元の推定に特に効果的であることがわかった。

係り元の推定が改善された事例においては、機能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、「として」の場合、構成要素である形態素列を独立に扱うと、図9(a)の例文において、「チェチェン進行を」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「して」という文節に係ってしまう。それに対して、「として」を機能表現として扱った場合、図9(b)のように、「チェチェン進行を」の係り先を正しく推定することができる。

また、係り元の推定が改悪された事例においては、機能表現の検出ミスが改悪の主な原因であった。

一方、係り先の推定が改善された事例においては、機能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、「として」の場合、構成要素である形態素列を独立に扱うと、図10(a)のように構成要素の一つである動詞「する」の連用形「し」が、最も近くの動詞と並立に係ると誤判定されることがある。それに対して、「として」を機能表現として扱った場合、図10(b)のように係り先を正しく判定できる。

逆に、機能表現を考慮した係り受け解析によって、係り先の推定精度があまり改善されない原因としては、内容的用法と機能的用法とで、係り先の特徴が変化する表現がほとんどないと

・・・| ロシア 軍 の | チェチェン 進行 を | 東欧 諸国 の | 首脳 と | して | ・・・| 批判 。

(a) ベースラインによる失敗例

・・・| ロシア 軍 の | チェチェン 進行 を | 東欧 諸国 の | 首脳 として | ・・・| 批判 。

(b) 提案手法による成功例

図9 係り元同定の改善例（助詞型一連用辞類）

チャンピオンと | して | つらい | 思い の | ときに | 出合った の が | ・・・

(a) ベースラインによる失敗例

チャンピオンとして | つらい | 思い の | ときに | 出合った の が | ・・・

(b) 提案手法による成功例

図10 係り先同定の改善例（助動詞型）

⁵ 提案手法（検出器出力使用）の係り元精度 0.740(= $\frac{5251}{7096}$) およびベースライン（CaboCha（機能表現抜き）の係り元精度 0.725(= $\frac{5148}{7100}$) の母比率の差の検定による。

ということが挙げられる。例えば、『「絶対に勝つ」という自信満々な人もいた。』という文章において、「という」は内容的に働いており、その係り先は「人も」という文節である。また、『トップという名にこだわる人もいる。』という文章において、「という」は機能的に働いており、その係り先は「名に」という文節である。この様に、「という」は内容的用法であっても機能的用法であっても、名詞を含む文節に係る特徴がある。

機能表現候補が内容的用法・機能的用法のいずれであるかということは、上で述べた通り、係り先の推定精度の改善にはあまり寄与しない。しかし、機能表現の係り先は、機能表現の品詞分類に依存する傾向がある。例えば、連用辞類の「として」は、動詞を含む文節に係るという特徴をもっているが、連体辞類の「という」は、動詞を含む文節には係らず、名詞を含む文節に係るという特徴を持っている。提案手法では、機能表現の品詞分類を行っており、機能表現の品詞を、相当する既存の品詞の細分類として扱うことによって、この問題を解決している。それに対して、CaboCha（オリジナル）では、全ての機能表現に対して、「助詞一格助詞一連語」という品詞を与え、機能表現の品詞分類を全く行っていない。このことが原因で、CaboCha（オリジナル）の係り先精度が、CaboCha（機能表現抜き）の係り先精度を下回っていると考えられる。

5 関連研究

(内元, 高岡, 野畑, 山田, 関根, 井佐原 2004; Uchimoto, Takaoka, Nobata, Yamada, Sekine, and Isahara 2004) は、話し言葉コーパス (前川 2004) を対象コーパスとして、半自動で精度良く短単位・長単位の2種類の粒度の形態論的情報を付与する枠組みを提案している。この枠組みでは、なるべく少ない人的コストで話し言葉コーパス全体に2種類の粒度の形態素情報を付与するため、最初に短単位の解析を行い、次に、短単位の形態素情報を素性として、短単位をチャンキングすることによって長単位の形態素情報を付与するという手順を採っている。例えば、「という」という機能表現は、短単位列としては助詞「と」および動詞「いう」の連体形の2短単位に分割され、長単位としては助詞「という」という1長単位にチャンキングされる。短単位から長単位をチャンキングするための機械学習手法としては、最大エントロピー法 (ME) と SVM を比較し、SVM がより優れていると報告している。内元らの研究は、話し言葉コーパス全体を対象としているのに対して、本論文では、機能表現に焦点をあてて検討を行っている点で異なる。そのため、内元らは話し言葉コーパス中の長単位全体に対する形態素解析精度の評価を行っているが、機能表現に特化した評価は行っていない。一方、本論文では、既存の解析系における機能表現の取り扱い状況を整理した上で、機能表現に特化した性能評価を行っている。また、本論文では、対象となる機能表現のリストを事前に用意しているため、形態素列のどの部分が機能表現として検出される可能性があるかという情報（チャンク素性およびチャン

ク文脈素性)を利用して, チャンキングを行うことができる. 機械学習手法としては, CRF と SVM を比較し, SVM の方が検出性能が高いことを示している.

(Shudo, Narahara, and Yoshida 1980; 首藤, 吉村, 武内, 津田 1988; 首藤, 小山, 高橋, 吉村 1998; Shudo, Tanabe, Takahashi, and Yoshimura 2004) は, 機能表現や慣用表現を含む複数の形態素からなる定型的表現をできるだけ網羅的に収集し, 機能表現間に類似度を定義して, 機能表現の言い換えや機械翻訳に利用することを提案している. (兵藤, 若田, 池田 1998; 兵藤, 池田 1999; 兵藤, 村上, 池田 2000) と (伊佐治, 山田, 池田 2004) は, 日本語の文構造の解析を容易にするため, 通常よりかなり長い文節を単位として解析を行うことを提案し, 機能表現を含む大規模な長単位機能語辞書を作成している. しかし, これらの先行研究における日本語処理系においては, 機能表現と同一の形態素列が内容的に振る舞う可能性が考慮されていない.

(黒橋, 河原 2005b) と (工藤, 松本 2002b) は, 機能表現を考慮して, 係り受け解析を実現している. (黒橋, 河原 2005b) では, 接続詞として形態素解析辞書に登録されている機能表現は, 形態素解析時に検出される. 次に, 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, 直前の文節の一部にまとめたり, 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, 機能表現を考慮した係り受け解析を実現している. (工藤, 松本 2002b) では, 形態素解析辞書に「助詞・格助詞・連語」や「接続詞」として登録されている機能表現は, 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ることによって, 機能表現を考慮した係り受け解析を実現している. しかし, 3.2.3 節で述べた通り, これらの手法において考慮されている機能表現の数は, 我々の一連の研究において対象とした機能表現の数よりも少ない. また, これらの研究では, 機能表現検出が係り受け解析にどれだけ効果的かという評価を行っていない. 一方, 本論文では, 評価対象を機能表現候補を含む文節に限定し, 機能表現検出が係り受け解析にどのような影響を与えるのかを調べ, 機能表現検出が, 係り受け解析に効果的であることを示している.

(土屋他 2007) では, 本論文の 3 節の内容に相当する機能表現のチャンキングについて述べられており, 本論文では, この結果をふまえて, 機能表現検出の結果を考慮した日本語係り受け解析手法 (4 節) を提案している. (土屋他 2007) と本論文との差分は 4 節の内容に相当するが, 技術的な内容を本論文の記述範囲で完結させるために, 本論文では, 3 節を設けて, 機能表現のチャンキングについても記述している.

6 結論

本論文では, 機能表現検出と形態素解析は独立に実行可能であると仮定した上で, 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, 機械学習手法を適用して機能表現の検出を実現し, さらに, その機能表現検出を利用して日本語機能表現を考慮した係

り受け解析を実現した。実際に、SVM を用いたチャンカー YamCha 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、59 種類の機能表現を対象として性能評価を行った。その結果、機械学習によって作成した機能表現検出器は、既存の解析系および人手で作成した規則を用いた検出器よりも、高精度に機能表現を検出できることを示した。係り受け解析に関しても、機能表現を考慮した訓練データから、係り受け解析・学習ツールを CaboCha を利用して学習を行い、機能表現検出器の解析結果を入力とする日本語機能表現を考慮した係り受け解析器を実装した。59 種類の機能表現を対象とした評価実験において、総体的に従来の CaboCha よりもよい性能を示すことができた。今後の研究課題として、対象とする機能表現の種類を増やし、その性能を評価することを計画している。また、格解析との統合的解析の実現により、解析性能をさらに改善することが期待できると考えている。

参考文献

- 浅原正幸, 松本裕治 (2003). “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>.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
- 兵藤安昭, 池田尚志 (1999). “文節単位のコストに基づく日本語文節解析システム.” 言語処理学会第 5 回年次大会発表論文集, pp. 502–504.
- 兵藤安昭, 村上裕, 池田尚志 (2000). “文節解析のための長単位機能語辞書.” 言語処理学会第 6 回年次大会発表論文集, pp. 407–410.
- 兵藤安昭, 若田光敏, 池田尚志 (1998). “文節ブロック間規則による浅い係り受け解析と精度評価.” 電子情報通信学会研究報告, NLC98-30 巻.
- 伊佐治和哉, 山田将之, 池田尚志 (2004). “長単位の機能語を辞書に持たせた文節構造解析システム ibukiC.” 言語処理学会第 10 回年次大会発表論文集, pp. 636–639.
- 国立国語研究所 (2001). 現代語複合辞用例集.
- Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). “Applying Conditional Random Fields to Japanese Morphological Analysis.” In *Proceedings of EMNLP*, pp. 230–237.
- 工藤拓, 松本裕治 (2002a). “Support Vector Machine を用いた Chunk 同定.” 自然言語処理, 9 (5), pp. 3–21.
- 工藤拓, 松本裕治 (2002b). “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, 43 (6), pp. 1834–1842.
- Kudoh, T. (2000). “TinySVM: Support Vector Machines.” <http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/index.html>.

- 黒橋禎夫, 河原大輔 (2005a). 日本語形態素解析システム JUMAN version 5.1 使用説明書. <http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman/juman-5.1.tar.gz>.
- 黒橋禎夫, 河原大輔 (2005b). 日本語構文解析システム KNP version 2.0 使用説明書. <http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp/knp-2.0.tar.gz>.
- 黒橋禎夫, 長尾眞 (1997). “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.
- Lafferty, J., McCallum, A., and Pereira, F. (2001). “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.” In *Proceedings of ICML*, pp. 282–289.
- 前川喜久雄 (2004). 『日本語話し言葉コーパス』の概観 ver.1.0. http://www2.kokken.go.jp/~csj/public/members_only/manuals/overview10.pdf.
- 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 (2003). “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>.
- 松吉俊, 佐藤理史, 宇津呂武仁 (2005). “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.
- 松吉俊, 佐藤理史, 宇津呂武仁 (2006). “階層構造による日本語機能表現の分類.” 言語処理学会第 12 回年次大会論文集, pp. 408–411.
- 森田良行, 松本正恵 (1989). 日本語表現文型, NAFL 選書, 5 巻. アルク.
- 中塚裕之, 佐藤理史, 宇津呂武仁 (2005). “助動詞型機能表現の形態・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.
- Shudo, K., Narahara, T., and Yoshida, S. (1980). “Morphological Aspect of Japanese Language Processing.” In *Proceedings of the 8th International Conference on Computational Linguistics (COLING'80)*, pp. 1–8.
- 首藤公昭, 小山泰男, 高橋雅仁, 吉村賢治 (1998). “依存構造に基づく言語表現の意味的類似度.” 電子情報通信学会研究報告, NLC98-30 巻, pp. 33–40.
- Shudo, K., Tanabe, T., Takahashi, M., and Yoshimura, K. (2004). “MWEs as Non-propositional Content Indicators.” In *Proceedings of the 2nd ACL Workshop on Multiword Expressions: Integrating Processing (MWE-2004)*, pp. 32–39.
- 首藤公昭, 吉村賢治, 武内美津乃, 津田健蔵 (1988). “日本語の慣用的表現について—語の非標準的用法からのアプローチ—.” 情報処理学会研究報告, 1988-NL-66 巻, pp. 1–7.
- Tjong Kim Sang, E. (2000). “Noun Phrase Recognition by System Combination.” In *Proceedings of the 1st Conference of NAACL*, pp. 50–55.
- 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 (2007). “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, 14 (1), pp. 111–138.

- 土屋雅稔, 宇津呂武仁, 松吉俊, 佐藤理史, 中川聖一 (2006). “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, **47** (6).
- 土屋雅稔, 宇津呂武仁, 佐藤理史, 中川聖一 (2005). “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.
- Uchimoto, K., Takaoka, K., Nobata, C., Yamada, A., Sekine, S., and Isahara, H. (2004). “Morphological Analysis of the Corpus of Spontaneous Japanese.” *IEEE Transactions on Speech and Audio Processing*, **12** (4).
- 内元清貴, 高岡一馬, 野畑周, 山田篤, 関根聡, 井佐原均 (2004). “『日本語話し言葉コーパス』への形態素情報付与.” 第 3 回「話し言葉の科学と工学」ワークショップ論文集, pp. 39–46.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

略歴

- 注連 隆夫**：2005 年大阪府立大学工学部卒業。2007 年京都大学大学院情報学研究科修士課程修了。現在、日本電気株式会社 C&C イノベーション研究所勤務。在学中は自然言語処理の研究に従事。
- 土屋 雅稔**：1998 年京都大学工学部 電気工学科第二学科 卒業。2004 年京都大学大学院 情報学研究科 知能情報学専攻 博士後期課程単位認定退学。2004 年より豊橋技術科学大学 情報メディア基盤センター 助手。2007 年より同助教。京都大学博士（情報学）。自然言語処理に関する研究に従事。
- 松吉 俊**：2003 年京都大学理学部卒業。2005 年同大学院情報学研究科修士課程修了。現在、同大学院情報学研究科博士後期課程在学中。自然言語処理の研究に従事。
- 宇津呂武仁**：1989 年京都大学工学部 電気工学第二学科 卒業。1994 年同大学大学院工学研究科 博士課程電気工学第二専攻 修了。京都大学博士（工学）。奈良先端科学技術大学院大学 情報科学研究科 助手、豊橋技術科学大学 工学部 情報工学系 講師、京都大学 情報学研究科 知能情報学専攻 講師を経て、2006 年より筑波大学 大学院システム情報工学研究科 知能機能システム専攻 助教。2007 年より同准教授。自然言語処理の研究に従事。
- 佐藤 理史**：1983 年京都大学工学部電気工学第二学科卒業。1988 年同大学院工学研究科博士後期課程電気工学第二専攻研究指導認定退学。京都大学工学部 助手、北陸先端科学技術大学院大学情報科学研究科助教授、京都大学大学院情報学研究科助教授を経て、2005 年より名古屋大学大学院工学研究科電子情報システム専攻教授。工学博士。自然言語処理、情報の自動編集等の研究に従事。

注連，土屋，松吉，宇津呂，佐藤

日本語機能表現の自動検出と統計的係り受け解析への応用

(2007 年 4 月 14 日 受付)
(2007 年 6 月 7 日 再受付)
(2007 年 7 月 10 日 採録)