話し言葉の係り受け解析と文境界推定の相互作用による高精度化

下岡 和也 † **内元 清貴** ‡ **河原 達也** † **井佐原 均** ‡

† 京都大学情報学研究科

‡ 独立行政法人通信総合研究所

e-mail: shitaoka@ar.media.kyoto-u.ac.jp

あらまし

「日本語話し言葉コーパス (CSJ)」を対象として係り受け解析や重要文抽出、談話構造分析等の処理を行う際には、書き起こしを基本的な処理の単位である「文」に区切っておく必要がある。本稿では、CSJ を対象にして、係り受け情報を利用して文境界推定を行う手法、および機械学習を用いて文境界推定を行う手法について比較・検討する。その結果、従来の統計モデルのみを用いた手法に比べて、文境界推定精度が改善し、F 値で最大 84.56 となった。さらに、文境界推定精度が向上することにより、係り受け解析精度も 75.21%から 77.27%に改善できることを確認した。

キーワード 話し言葉, 係り受け解析, 文境界推定, 機械学習

Interaction of Dependency Structure Analysis and Sentence Boundary Detection in Spontaneous Japanese

Kazuya Shitaoka† **Kiyotaka Uchimoto**‡

Tatsuya Kawahara† **Hitoshi Isawara**‡

† School of Informatics, Kyoto University

‡ Communications Research Laboratory

e-mail: shitaoka@ar.media.kyoto-u.ac.jp

Abstract

For dependency structure analysis, extraction of key sentences and discourse structure analysis, it is necessary to divide text into the “sentence” unit. However, sentence segmentation is not straightforward for transcription of spontaneous speech. We present a method of sentence boundary detection for the Corpus of Spontaneous Japanese by using the dependency structure and also investigate machine learning techniques. Experimental results show that the accuracy of sentence segmentation is improved with these methods, and also that the accuracy of dependency structure analysis is improved by using the enhanced sentence boundary detection.

Key words spontaneous speech, dependency structure analysis, sentence boundary detection, machine learning

1　はじめに

「日本語話し言葉コーパス (CSJ)」[1] を対象として、重要文抽出 [2, 3]、係り受け解析、談話構造分析 [4] 等の様々な研究が行われるようになった。これまでの日本語の自動係り受け解析の研究 [5, 6, 7, 8, 9] の多くは書き言葉を対象としたものである。一方、話し言葉では句点が明示的でなく、CSJ にも句点は書き起こされていないため、係り受けをすべての文節に対して特定しようとすると、文間関係も文節の関係として定義する必要が生じる。しかし、文間関係の特定は難しく、人による揺れが顕著である。

また、自動要約の際に圧縮をしたり、格関係などを抽出する目的のためには文単位の係り受けで十分であるため、実際には文内の文節間係り受けを対象とした処理が行われている。このように、係り受け解析に限らず、重要文抽出による要約、談話構造分析等の様々な処理を行う際の基本的な処理の単位は「文」であるが、話し言葉においては書き言葉ほど「文」の定義が明確ではない。したがって、あらかじめ話し言葉における「文」を定義し、これらの処理を行う前段階として書き起こしを「文」に区切っておくことが望ましい。CSJ では、「節」を採用することで独自に話し言葉における「文」の単位を定義している。

このような「文」を自動検出する方法として、あらかじめ用意した節パタンを用いたパタンマッチングによる方法が提案されている [10]。しかし、パタンマッチングでは体言止めや倒置など話し言葉特有の問題に適切に対処することができず、実際の文境界のタグ付与は人手による修正を必要としている。我々もこれまで自動で文境界推定をする研究を行ってきた [11]。しかし、この手法では文境界候補にポーズの存在を仮定し、文字列のパタンマッチングで検出しているため、文境界の全てを候補として検出することは難しく、推定された結果は再現率が十分でないという問題があった。

そこで本稿では、係り受けの情報を新たに文境界推定に用いることを考える。実際に人手により文境界を検出する際にも係り受けの情報は用いられている。また従来手法は直接的に文境界の検出の学習を行っていなかったが、ここでは機械学習の手法として SVM を用いることも比較・検討する。さらに、提案手法による文境界推定で得られた結果を用いて係り受け解析の精度を上げることも併せて考える。

2　話し言葉の係り受け解析と文境界推定

話し言葉は書き言葉と大きく異なる。そのため、話し言葉を対象として係り受け解析・文境界推定を行う際には、書き言葉では見られない話し言葉特有の問題が生じる。以下では、それぞれの処理を行う際に生じる話し言葉特有の問題について説明し、それに対してどのように対処するかについて述べる。

2.1　係り受け解析における問題点

文境界が明示されていない

話し言葉では文境界が明示されていない。そのため、全ての文節に対して係り受けを特定する際には、文間関係も文節の関係として特定する必要がある。しかし、予備実験で複数の被験者に係り受けを付与してもらったところ、文間関係に相当する係り受けは、被験者間の揺れが大きく、安定して係り受けを特定するのが難しいことがわかった。また、自動要約のための文圧縮において不要な要素を削除する場合などで実際に必要となるのは、文節間の修飾・被修飾関係や述語と格要素の関係といった文内の係り受け関係であることが多い。したがって、本研究では、文内の文節間係り受けを対象とし、同時に、文境界の推定も行なう。文境界の定義は次節で述べる CSJ のものに従う。

この問題が話し言葉の係り受け解析を困難にする最も大きな問題と考え、本稿では主にこの問題に着目しその対処法について述べる。

係り先がない文節がある

話し言葉では、途中で発話のプランが変わったために係り先が消失したり、「あのー」「そのー」などのフィラー、フィラー的な振る舞いをする副詞の「もう」、「はい」「うん」などの相槌、文頭の接続詞「で」、言いよどみなどのように、係り受け関係を特定しても用途はほとんど考えられず、係り受けを定義することに意味がない場合などがある。このような場合、CSJ における定義では係り受けを付与していない。

フィラーや相槌、言いよどみについては、話し言葉に数多く出現するので無視できないが、例

えば、浅原らの手法 [12] を用いて係り受け解析の前にある程度特定できると考えており、本稿ではすべて削除して扱う。これらを削除するにあたり、本研究では形態素の品詞情報を用いることにした。フィラーや相槌には「感動詞」・言いよどみには「言いよどみ」のタグが付与されている。

それ以外の係り先を持たない文節については、便宜上、すべて直後の文節に係るものとして扱う。これらに関しては、本来、正しく「係り先なし」と推定するべきであるが、その推定については今後の課題とする。

係り受け関係が交差する

一般に、日本語の書き言葉においては「係り受け関係は互いに交差しない」という非交差条件が成り立つと言われている。しかし、話し言葉ではこの非交差条件が成り立たないことも多い。例えば「これが 私は 正しいと 思う」といった場合、「これが」が「正しいと」に係り、「私は」が「思う」に係るので係り受け関係が交差している。しかし、実際に CSJ においても交差している文節の数はそれほど多くないため、本稿では、係り受けの非交差条件が成り立つと仮定して係り受け解析を行う。交差している場合については今後の課題である。

言い直しが多い

話し言葉ではしばしば言い直しが生じる。CSJ では、言い直し関係には、係り受け関係と同様の関係が付与され、さらに、D というラベルが付与されている。言い直し関係以外にも、並列関係・同格関係も係り受け関係と同様の関係が付与され、さらに、それぞれ P・A というラベルが付与されている。このうち、並列関係・同格関係については、書き言葉のコーパスである「京大コーパス」の基準に準拠している。本来は、文節間の関係の推定のみではなくそれがどういった関係なのかまで推定すべきであるが、書き言葉を対象にした研究においても多くの場合は関係の有無の推定のみを対象としているため、同様にする。

倒置表現がある

話し言葉ではしばしば倒置表現が用いられる。CSJ では、倒置は左係りで表現されている。本稿では、関係を特定することが重要と考え、CSJ における倒置に対して人手で修正を加え、便宜上、すべて右係りになるようにして用いた。具体的には、「ずっと 待ってるんですよ 大の 男が」という文に対して、実際には「男が」が「待ってるんですよ」に倒置で係るわけだが、「待ってるんですよ」が「男が」に係るように修正した。

2.2 文境界推定における問題点

日本語の話し言葉においては、文の定義が明確でない。CSJ では、節境界を文境界候補とすることで独自に話し言葉における「文」の単位を定義している [10]。

まず、CSJ では節境界として次の 3 種類を定義している。

絶対境界 : これはいわゆる文末表現で、述語の終止形・終助詞・「と文末」など。

強境界 : 並列節「ケレドモ」「ガ」「シ」・「ましテ」節・「でしテ」節など。

弱境界 : 理由節「カラ」「ノデ」・連用節・引用節・条件節「タラ」「ト」「ナラ」「レバ」など。

これらの節境界のうち、絶対境界と強境界は基本的に文境界となり、弱境界は機能的に区切れていると判断される箇所のみが文境界となる。この判断は人手により行われるのだが、その際に弱境界に対する処理と並行して話し言葉特有の現象である「体言止め」や「言いさし」・「倒置」などの箇所に対する修正も行われる。本稿では、以上のような処理を経て検出された箇所を文境界として用いる。

我々が従来行ってきた文境界推定 [11] では、主に上記の絶対境界にあたる箇所を対象としており、文境界候補にポーズ長を含めた文字列のパタンマッチングで検出してきた。しかし、上記 3 種類の節境界のうち、絶対境界以外は直後にポーズが置かれないことも多く、また、体言止めなどをパタンマッチングで検出するのは困難である。

そこで本稿では、人手による文境界検出の際にも用いられている係り受け情報を新たに利用する。また、さらに精度を向上させるため機械学習に基づく手法を導入する。

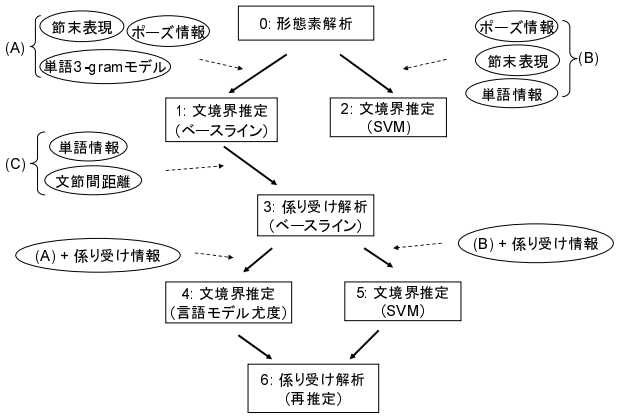

図 1: 係り受け解析・文境界推定の概要

3 係り受け解析と文境界推定のアプローチ

図 1 に本稿で行う処理の概要を示す。以下では、係り受け解析・文境界推定のそれぞれについて本稿でどのように処理しているかを説明する。

3.1 係り受け解析

統計的日本語係り受け解析では、二文節間の係りやすさは確率値で表される。この確率値は係り受け確率モデルから求められる。

文全体の係り受け関係 $\mathbf{D}$ がそれぞれの文節 $b_i (i = 1, \ldots, n-1)$ を係り元の文節とする係り受け関係 D_i の順序付き集合 $\mathbf{D} = \{D_1, \ldots, D_{n-1}\}$ で表されると仮定すると、統計的係り受け解析とは、入力文 S が与えられたときに文全体の係り受けが $\mathbf{D}$ となる確率 $P(\mathbf{D}|S)$ が最も高くなるような全体の係り受け関係 $\mathbf{D}_{best}$ を求める処理のことである。日本語の係り受けには，主に以下の特徴があるとされており、これらの特徴を満たすような $\mathbf{D}_{best}$ を求める。

1: 係り受けは前方から後方に向いている
2: 係り受け関係は交差しない
3: 係り要素は受け要素を一つだけ持つ

従来用いられている係り受け解析モデル [5, 8, 9, 13] では、二つの文節の関係を「係る」か「係らない」かの二カテゴリとして学習し、基本的には、着目している二文節間の関係のみを考慮して二文節が係る確率を求めている。

しかし、本稿で用いる係り受け解析モデル [6] では、二つの文節間の関係を「間」「係る」「越える」の三カテゴリとして学習し、着目している二文節の間にある文節や、それらよりも文末側にある文節との関係も考慮して二文節が係る確率を求めている。そのため、従来のモデルに比べてより多くの情報を考慮できると考えられる。

このモデルを最大エントロピー (ME) モデルとして実装した。ME に与える素性としては、単語の表層表現・品詞・活用形、文節間距離 (およびそれらの組合せなど) を利用している。また、$\mathbf{D}_{best}$ を求めるために、文末から文頭に向けて解析することにより，効率良く組み合わせの数を減らしながら一文全体の係り受けを決定する方法を用いている。この方法では解の探索をビームサーチにより行っているが、決定的に解析を行ってもビーム幅を広くしたときとほとんど同じ精度が得られることが実験によりわかっている。したがって、本稿でも文末から決定的に解析する。

3.2 言語尤度を利用した文境界推定 (従来手法)

これまで、我々は自動で文境界を推定する研究を行ってきた [11]。ここではその手法について説明する。

句点を含まないがポーズ情報を含む文字列 X と、句点を含む文字列 Y を別の言語と考え、統計的機械翻訳により、式 (4) に示すように $P(Y|X)$ を最大にする文字列 Y を求める問題として定式化する。具体的には、ポーズが句点に変換されうる ($P(X|Y) = 1$ となる) 全ての箇所に対して、句点を挿入する場合としない場合の言語モデル尤度 $P(Y)$ を比較し、句点挿入の判定を行う。

$$\max_Y P(Y|X) = \max_Y P(Y)P(X|Y) \qquad (1)$$

変換モデル $P(X|Y)$ には、ポーズ前後の表現とポーズ長に依存するモデルを用いる。ポーズ前後の表現として 2.2 節で述べた節境界の表現を用いた。話し言葉特有の文末表現「〜と」「〜ない」「で〜」、および文末以外でも頻繁に用いられる文末表現「〜た」においては平均ポーズ長以上の場合のみ挿入しうるとし、それ以外の表現では短いポーズ長でも挿入しうるとした。言語モデル尤度 $P(Y)$ の計算には、文境界が人手により付与された CSJ の書き起こしから学習された単語 3-gram モデルを用いる。

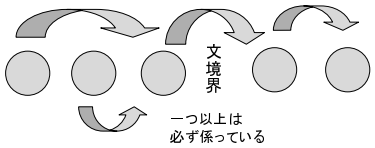

図 2: 自分に係る文節が 1 つ以上ある

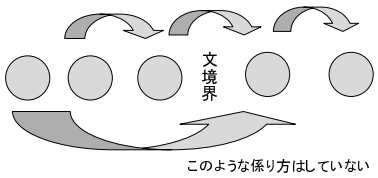

図 3: 自分を飛び越える文節がない

3.3 係り受け情報を用いた文境界推定

我々が提案する係り受けを用いた文境界推定の手法について説明する。従来手法では、句点が挿入されうる箇所の候補をポーズ長を含めた文字列のパタンマッチングで行ったきた。ここでは挿入されうる箇所の候補を、係り受け情報を用いて検出することを考える。

文境界直前の文節の係り受け関係について次の 3 つが考えられる。

(1) 自分に係る文節が 1 つ以上ある (図 2)

係り受け関係は 1 つの文内で閉じている。そのため、倒置文を除いては文境界直前の文節にはその 1 つ前の文節が必ず係ることになる。また本稿では、2.1 節で述べたように倒置で係っている箇所も人手で右係りになるように修正しているため、倒置文においても同様のことが成り立つ。よって、文境界直前の文節には必ず 1 つ以上自分に係る文節が存在すると仮定できる。

(2) 自分を飛び越える文節がない (図 3)

(1) と同様、係り受け関係は 1 つの文内で閉じていることから、係り受け関係が文をまたぐことは原則としてない。よって、文境界直前の文節には自分を飛び越えて係る文節は存在しないと仮定する。

(3) 自分が係る確率は低い (図 4)

本稿では、文境界直前の文節は便宜上直後の文節に係るものとして扱っているが、本来これらの文節はどこにも係らないものである。よって、3.1 節で述べた手法を用いて得られた係り受け解析結果において、文境界直前の文節が係り先の文節に係る確率は低いものになっていると考えられる。

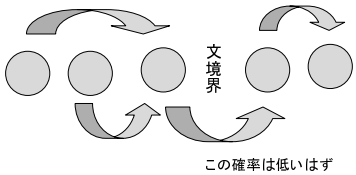

図 4: 自分が係る確率は低い

以上 (1)〜(3) の条件を全て満たす文節を、文境界直前の文節候補として検する。そして、3.2 節において検出される文境界候補と統合した後、同様に言語モデル尤度により句点挿入判定を行う。なお、(2) については、今回扱った 188 講演全てにおいて 50 文節以上離れて係る場合がなかったため、係り受け解析結果において 50 文節以上離れて係っている文節は無視することにした。また、あまりに低い確率で係っている文節については、その係り先が誤っている可能性が高いため、これも無視することにした。この閾値となる確率をパラメータ p とする。同様に、(3) における閾値となる確率をパラメータ q とする。予備実験により、最適なパラメータ (p, q) を求める。

本手法により得られる文境界直前の文節候補数は全文節数のおよそ 1/3 程度になる。そのため、前後 2 単語内に文境界候補となる別の単語が含まれていることが十分に考えられる。したがって、前後 2 単語内に文境界候補となる表現が存在しなくなる範囲において、その全ての変換パターンの尤度を比較して最終的な出力を決定することとした。場合によっては相当数の変換パターンが生成される可能性もあるので、ビームサーチを導入した。なお、ビーム幅は 10 で固定した。

3.4 機械学習を利用した文境界推定

次に、文境界推定において、機械学習を利用することを考える。本稿では、Suport Vector Machine(SVM) を用いることとした。SVM は 2 クラスの分類を行う機械学習アルゴリズムである。

なお、本稿では文境界推定の問題をテキストチャンキングの問題として扱う。テキストチャンカーとして

は、SVM ベースの YamCha[7] を用いることとした。YamCha では、カーネル関数には多項式カーネルが用いられており、現在位置の単語のチャンクタグを推定する際に、前後 2 単語の単語情報を静的素性として、推定により得られた前 2 単語のチャンクタグを動的素性として用いている。また、解析方向を逆にすることで後ろ 2 単語のチャンクタグを素性として用いることも考えられている。さらに、多値クラスの識別問題に対処するため、pairwise classification と呼ばれる手法を用いている。これは、N クラスの識別問題を解くために、各クラス対の組合せを識別する $N*(N-1)/2$ 種類の識別器を作成し、最終的にそれらの多数決で決定する手法である。

4 実験と評価

ここでは、係り受け解析と文境界推定の実験結果と考察について述べる。実験に用いたコーパスは CSJ の 188 講演の書き起こしである。テストデータは、全ての実験を通じて同一の 10 講演を用いた。なお、係り受け解析については、講演の最後の文節を除く残り全ての文節に対して係り先を正しく推定できた文節の割合 (係り受け正解率) で、文境界推定については F 値で評価を行う。また、係り受け解析を行う際にはテストデータが closed な場合と open な場合の 2 通りを行い精度を比較する。以下の表 1〜3 において、それぞれの場合を (closed な場合)・(open な場合) と表記する。

まず、係り受け解析および文境界推定精度のベースラインを求めた。文境界推定のベースラインの手法としては 3.2 節で述べた手法を用いた (図 1 の処理 1)。$P(Y)$ の推定に使用する言語モデルはテストデータを除く 178 講演で学習した。その結果、再現率 64.51%、適合率 94.17%、F 値 75.57 であった。次に、係り受け解析のベースラインについて述べる。全く文境界が示されていないデータに対して open テストで係り受け解析を行ったところ、文境界直前の文節を評価の対象外とした場合で 74.58%であった。しかし、本稿では文内の文節間係り受けを対象としているため、文境界直前の文節も評価の対象とし、そこを文境界と正しく推定できているかを含めて係り受け解析の評価を行う必要がある。そこで、上記のベースライン手法により文境界を推定したデータに対して係り受け解析を行った結果をベースラインとした (図 1 の処理 3)。結果は、open テストで 75.21%、closed テストで 80.74%であった。

なお、文境界直前の文節は 3.3 節で述べたように直後の文節に係るとして扱っているが、上記の係り受け解析ではその確率は求まらないように実装されている。したがって、以下の実験では文境界直前の文節が係る確率を、他の文節との整合性を考えて全て 0.5 と固定することとした。

4.1 係り受け解析結果を用いた文境界推定

ベースラインの係り受け解析の結果を用いて、3.3 節で述べた手法により文境界推定を行った (図 1 の処理 4)。

まず、3.3 節で述べた 2 つのパラメータ (p, q) についてチューニングを行った。チューニングデータとしてテストデータと異なる 15 講演を用いた。その結果、係り受け解析が open テストの場合、$(p, q) = (0, 0.9)$ の時に F 値 78.34、係り受け解析が closed テストの場合、$(p, q) = (0, 0.8)$ の時に F 値 78.59 で最大となった。よって、以降の実験ではこの値を用いる。ともに $p = 0$ なので、自分を飛び越える文節について無視する条件は、「50 文節以上離れて係る」ということのみで、それ以外はすべて考慮することとなる。

得られた結果を表 1 に示す。係り受け解析が open な場合は F 値で約 1.4 程度、closed な場合は F 値で 2.0 上昇した。用いた係り受け解析の精度が、上記にあるように open な場合と closed な場合とで約 5.5%異なるにもかかわらず、文境界精度が約 0.5 しか違わない。これは、文境界と関係している係り受けに関しては open な場合でも closed な場合と同等の精度が得られているためと考えられる。

言語モデル尤度による判定を行わずに、検出された候補を全て文境界であるとした場合の精度は、ベースラインの手法では再現率 68.23%(769/1127)、適合率 81.54%(769/943)、提案手法で係り受け解析が open な場合では再現率 87.22%(983/1127)、適合率 27.74%(983/3544) であった。つまり、係り受け情報を使うことで新たに 214 箇所を正しく文境界候補として検出できている。しかし、言語モデル尤度による判定を行った結果、これらのうち 108 箇所しか選ばれていない。選ばれていない箇所を調べてみると、体言止めの箇所、「〜と思う」「〜は難しい」といった動詞や形容詞で終わっている箇所、あるいは「〜というのは」「〜としては」といった箇所であった。

表 1: 係り受け情報を用いた文境界精度

	再現率	適合率	F 値
係り受け情報利用 (open な場合)	74.09% (835/1127)	82.51% (835/1012)	78.01
係り受け情報利用 (closed な場合)	74.18% (836/1127)	83.52% (836/1001)	78.57
ベースライン	64.51% (727/1127)	94.17% (727/772)	76.57

表 2: SVM を用いた文境界精度

	再現率	適合率	F 値
係り受け情報利用 (open な場合)	79.95% (901/1127)	89.74% (901/1004)	84.56
係り受け情報利用 (closed な場合)	79.59% (897/1127)	90.51% (897/991)	84.70
係り受け情報 利用せず	79.33% (894/1127)	90.12% (894/992)	84.38

これらの箇所、特に体言止めの箇所以外については、言語モデルの学習コーパスを増加することである程度対処できると考えられる。

一方、ベースラインの場合・係り受け情報を用いた場合ともに、誤って文境界が挿入されている箇所の多くは「〜が」「〜まして」「〜けれども」あるいは「〜て」といった箇所である。最初の 3 つの箇所は 2.2 節における絶対境界の表現で、基本的には文境界となるが、人手により「つなぐための明確な理由がある」と判断された場合文境界にはならず、「〜て」は 2.2 節における弱境界の表現で、人手により区切れていると判断された場合のみ文境界となる。これらの表現は意味的な理解を含めて文境界かどうかを判断されるため、言語モデル尤度のみでは正確に判定するのは困難である。

4.2 SVM を用いた文境界推定

次に、SVM を用いて文境界推定を行った (図 1 の処理 5)。学習にはテストデータを除く 178 講演を用いた。SVM に与える素性としては以下のものを用いた。なお、比較のため係り受け情報 (4)(5) を使用しない場合 (図 1 の処理 2) の精度も評価した。

(1) 前後 3 単語の単語情報 (表層表現・読み・品詞情報・活用の種類・活用形)
(2) 1 講演で正規化したポーズ長
(3) どの節境界候補にマッチしたか
(4) 自分が係る確率
(5) 自分に係る文節の個数とその確率

また、YamCha における具体的なパラメータは次のものとし、ラベリングスキームには IOE を用いた。

・多項式カーネルの次数：3
・解析方向：Left to Right
・多値クラス識別：Pairwise 法

表 2 に結果を示す。4.1 節の結果と比べて F 値が約 6.5 程度高い。これは、教師つき機械学習の効果と考えられる。しかし、係り受け情報を用いることの効果はほとんど見られず、再現率・適合率ともほとんど差がない。その原因として、係り受け解析において用いている素性と SVM で用いている素性が重複していることが考えれる。つまり、係り受け解析の際に素性として用いられている単語情報は SVM に与える素性 (1) とほとんど同じであるため、素性 (1) からすでに素性 (4)(5) の情報が得られているのではないかと考えられる。ただし、SVM では前後 3 単語しか見ていないため、それより離れた文節の係り受け情報は素性 (1) からでは得られない。それにもかかわらず精度が変わらない理由として、離れて係る文節と文境界があまり関係していない、あるいは離れて係る文節は精度が悪いため素性 (5) の情報が生かされていない、といったことが考えられる。

4.3 文境界推定結果を用いた係り受け解析

上記の 2 つの手法により得られた文境界推定の結果を用いて、再度、係り受け解析を行った (図 1 の処理 6)。表 3 にその結果を示す。4.2 節の結果は係り受け情報を用いた場合のものを使用した。open テスト、closed テストともに、最も精度が高かった文境界推定結果を用いることで約 2%程度上昇している。これは、文境界精度が上昇することでより多くの文末の文節を特定でき、また、それによって別の文の文節に誤って係っていた箇所などが改善されたためである。

ここで、文境界推定の影響を調べるため、精度が 100%であると仮定して実験を行った。結果は open テストで 80.59%、closed テストで 86.12%であった。つまり、完全に文境界が推定されたとしても、closed テストでさえ約 14%誤りがあり、書き言葉 (新聞記事) を対象とした場合よりも 8%近く精度が低い。これは、話し言葉には書き言葉のように読点がなく、また、挿入構造があるため離れた文節に係る場合も多いことなどが原因であると考えられる。

表 3: 文境界推定結果を用いた係り受けの再推定結果

	open な場合	closed な場合
4.1 節の結果	75.78%	81.20%
4.2 節の結果	77.27%	82.63%
ベースライン	75.21%	80.74%

5 まとめ

CSJ の講演を用いて話し言葉の係り受け解析および文境界推定を行った。話し言葉における「文」として、CSJ で定義されている単位を用いた。従来行ってきた文境界推定には用いていなかった係り受け情報を新たに利用し、また機械学習の手法として SVM を用いた。さらに、このように推定した文境界を用いることにより、係り受け解析も改善されることも検討した。

文境界推定については、従来手法による精度が F 値で 76.57 であるのに対し、係り受け情報を用いた場合で 78.01 となり、また SVM を用いることで 84.56 と改善された。係り受け解析については、従来手法による文境界推定結果を用いた場合 75.21%に対し、提案手法による推定結果 (SVM) を用いた場合で 77.27%となり、約 2%程度向上した。

今後の課題としては、実験の結果明らかになった問題点をふまえてさらに改善を図ることや、2.1 節で述べたように、今回対象としなかった話し言葉の係り受け解析における問題点に対処することなどが挙げられる。

参考文献

[1] 古井貞熙, 前川喜久雄, 井佐原均. 科学技術振興調整費開放的融合研究推進制度－大規模コーパスに基づく「話し言葉工学」の構築－. 日本音響学会誌, Vol.56, No.11, pp.752-755, 2000.

[2] 野畑周, 関根聡, 内元清貴, 井佐原均. 話し言葉コーパスにおける文の切り分けと重要文抽出. 「話し言葉の科学と工学」ワークショップ予稿集, pp. 93–100, 2002.

[3] 南條浩輝, 北出祐, 河原達也. 談話標識の統計的学習に基づいた講演からの重要文抽出. 日本音響学会研究発表会講演論文集, 2-6-18, 2003.

[4] 森本郁代, 高梨克也, 竹内和広, 小磯花絵, 井佐原均. 話し言葉コーパスへの談話構造タグ付与. 言語処理学会 第 9 回年次大会発表論分集, pp. 695–698, 2003.

[5] 内元清貴, 関根聡, 井佐原均. 最大エントロピー法に基づくモデルを用いた日本語係り受け解析. 情報処理学会論文誌, 36(10), pp. 2353–2361, 1999.

[6] 内元清貴, 村田真樹, 関根聡, 井佐原均. 後方文脈を考慮した係り受けモデル. 自然言語処理学会誌, Vol.7, No.5, 2000.

[7] T.Kudo and Y.Matsumoto. Chunking with support vector machines. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

[8] 藤尾正和, 松本裕治. 統計的手法を用いた係り受け解析. 情報処理学会 自然言語処理研究, NL117-12, pp. 83–90, 1997.

[9] 春野雅彦, 白井論, 大山芳史. 決定木を用いた日本語係り受け解析. 情報処理学会論文誌, 39(12), pp. 3177–3186, 1998.

[10] 高梨克也, 丸山岳彦, 内元清貴, 井佐原均. 話し言葉の文境界－ CSJ コーパスにおける文境界の定義と半自動認定－. 言語処理学会 第 9 回年次大会発表論分集, pp. 521–524, 2003.

[11] 下岡和也, 河原達也, 奥乃博. 講演の書き起こしに対する統計的手法を用いた文体の整形. 情報処理学会研究報告, SLP-41-3, 2002.

[12] 浅原正幸, 松本裕治. 形態素解析とチャンキングの組み合わせによるフィラー/言い直し検出. 言語処理学会第 9 回年次大会発表論文集, pp. 651–654, 2003.

[13] M Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 184–191, 1996.