話し言葉における引用節・挿入節の自動認定結果を利用した係り受け解析

浜辺 良二† 内元 清貴‡ 河原 達也† 井佐原 均‡
†京都大学大学院 情報学研究科 ‡独立行政法人 情報通信研究機構

1. はじめに

係り受け解析は日本語解析の重要な基本技術の一つとして認識されている．我々は,『日本語話し言葉コーパス (CSJ)[1]』に含まれる学会講演や模擬講演などのような，長い独話を対象として係り受け解析を行うことを考える．

話し言葉の係り受け解析は書き言葉に比べて困難である．その最大の原因は，句読点や括弧など，文章の節構造を明示する記号が含まれないことである．この問題に対して，文境界を自動推定するための手法が現在までに考案されてきた．本研究ではそれに加え，引用節や挿入節といった節構造を自動認定することを考え，SVM を用いたテキストチャンキングによる手法を提案する．さらに，既存の係り受け解析手法に，認定した引用節・挿入節の情報を取り入れることで，係り受け解析精度の向上を目指す．

2. 話し言葉の係り受け解析における問題点

話し言葉における係り受け解析では，書き言葉には見られない特有の問題が生じる．以下では，まず従来の問題点およびその対処法について述べる．さらに，本研究で取り上げる引用節・挿入節構造に関する問題点について説明する．

2.1 従来研究における問題点

(1) 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ことになる．しかし，文間関係については人間の判断が揺れる場合が多い．そこで本研究では，文間関係は推定せず，文境界を推定するに留める．係り受けは，文境界推定後，文内の文節間係り受けのみを対象として解析する．

(2) 係り先がない文節がある

話し言葉では，フィラーや言いよどみなど，係り受け関係を特定しても用途が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がある．このような場合，CSJ では係り受けが付与されていない．フィラーや言いよどみについては，浅原らの手法[2] を用いることである程度特定できると考え，本研究では全て削除して扱う．ただし，どこにフィラーがあったかについての情報は残しておき，後の解析に利用する．それ以外の係り先を持たない文節の扱いについては，3.1 節で述べる．

(3) 係り受け関係が交差する

(4) 言い直しが多い

(5) 倒置表現がある

上記 (3)(4)(5) の対処法については下岡らの手法[3] に従うものとする．

2.2 本研究で取り上げる問題点

本研究では，話し言葉の係り受け解析が困難である最大の要因は，節構造が明示されないことであると考える．その一例として，文境界が明示されていないことがこれまでに問題とされてきた．下岡らの研究[3] では，文末表現やポーズ長，係り受け情報を素性としたテキストチャンキングによって，文境界推定において F 値 84.9 を得ている．しかし，節構造に関して，さらに以下のような問題が挙げられる．

(6) 引用節が明示されない

引用節は，人の言ったことや思ったことを文に取り込む際に用いられる．書き言葉では，引用節には鍵括弧が付与されていることが多いが，話し言葉においてその範囲が明示されることはない．以下の例では, {} 内が引用節に相当する．

例）ここは┐
昔から┐
{一度でも┐
いいから┤
行ってみたい} と┤
思っていたところです

(7) 挿入節がある

話し言葉では，途中で発話のプランが変わることがある．その際に，前節で述べた言い直しや倒置に加え，以下のような挿入節構造が発生する場合がある．() 内が挿入節に相当する．

例）ホテルの┐
部屋の┐
中も┐
早速┐
（夜┐
着いたんですけども）┤
チェックしました

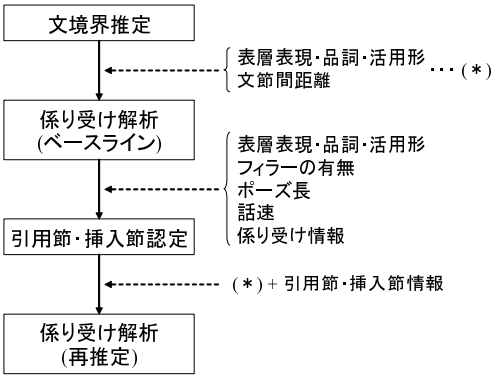

図 1 処理概要図

これら引用節・挿入節は，内部で係り受けが閉じているため，節末の文節を除いて，節の内部と外部の文節で係り受けを結ぶことはない．しかし，引用節・挿入節を含む文は節構造が複雑であるため，自動係り受け解析において，節の内部と外部で係り受けが結ばれてしまうことがある．これが，話し言葉の係り受け解析精度を悪くする一因となっている．

逆に，引用節・挿入節の区間を事前に得ることができれば，係り受け精度の向上が期待できる．そこで本研究では，引用節・挿入節の自動認定および，それを利用した係り受け解析手法を実現する．

3. 係り受け解析と引用節・挿入節の自動認定のアプローチ

図 1 に本手法で行う処理の概要を示す．以下では，係り受け解析および引用節・挿入節認定の手順についてそれぞれ説明する．

3.1 係り受け解析

本研究では，内元らの手法[4] に基づき，係り受け解析モデルを統計的に学習する．統計的係り受け解析では，文中の各文節がどの文節に係りやすいかを確率値で表し，それらを要素とした係り受け行列を作成する．そして，一文全体が最適な係り受け関係になるように，それぞれの係り受けを決定する．ここで，2 つの文節間の関係を「間」「係る」「越える」の 3 カテゴリとして学習することにより，着目している 2 文節の間にある文節や，それらより後方にある文節との関係も考慮して確率値を計算できる．

この係り受け解析モデルは最大エントロピー (ME) モデルとして実装され，素性には，単語の表層表現・品詞・活用形・文節間距離など（およびそれらの組合せ）が利用されている．本研究ではさらに，2 文節の間に引用節・挿入節の境界があるかどうかを素性に加えた．これにより，引用節・挿入節の情報を利用した係り受け解析法を実現した．2 文節の間に境界がある場合，この 2 文節の係る確率は低くなる．

表 1 チャンキングに使用するタグの種類

タグ	タグの説明
B	引用節／挿入節の始端
E	引用節／挿入節の終端
I	引用節／挿入節の内部 (始端，終端以外)
O	引用節／挿入節の外部
S	1 文節から成る引用節／挿入節

なお，係り先のない文節については，便宜上以下のように係り先を設定した．

- 引用節・挿入節中に文境界が含まれる場合，この文境界直前の文節は節末に係るとする．
- 引用節・挿入節末の文節が係り先を持たない場合，文末に係るとする．
- それ以外の係り先を持たない文節は，隣の文節に係るとする．

3.2 引用節・挿入節の認定

本研究では，引用節･挿入節の認定問題を文境界推定と同様にテキストチャンキングの問題として扱う．よって，引用節・挿入節の認定と文境界推定を同時に行うことが可能であり，これらを別々に行う場合に比べて，文境界推定の誤りに対しても頑健に動作することが期待できる．

テキストチャンカとして，SVM に基づく YamCha[5] を用いる．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，複数の素性の組合せを考慮した学習が可能である．また，推定により得られた前数文節のチャンクタグを動的素性として用いることができる．本手法では，多項式カーネル次数は 3，解析方向は Right to Left とし，後方 3 文節の動的素性を利用した．ラベルには，文境界に関するタグ (E: 文末, I: 文末以外) と，引用節／挿入節に関するタグ (表 1) の 3 つ組を用いる．以下にラベル付与の例を示す．

```
                                    (文境界, 引用節, 挿入節)
例) 今は──────────────┐ (I,O,O)
    ({予算の┐          │ (I,B,B)
      関係だ} と┐       │ (I,E,I)
        思いますが)      │ (I,O,E)
          一夏に──┐      │ (I,O,O)
            三回ぐらいしか──┤ (I,O,O)
                  やりません (E,O,O)
```

SVM に与える素性としては，以下のものを用いた．

(1) 単語情報 単語情報として，表層表現・読み・品詞情報・活用の種類・活用形を用いた．引用節の終端では「～と思う」「～って言う」などの表現が，挿入節の終端では「～ですが」「～けれども」などの表現が多用される．

(2) 文節前後のフィラー・ポーズ　引用節や挿入節の前後にはフィラーやポーズが入りやすいと考えられる．よって文節前後のフィラーの有無，ポーズ長を素性として利用した．なおポーズ長については，講演ごとに正規化した値を用いた．

(3) 文節の話速　挿入節では，話者が早口になると考えられるため，各文節の話速をポーズ長と同様に正規化してから用いた．

引用節・挿入節の終端を推定する際には単語情報が大きな手がかりとなるが，以上の素性は全て局所的な情報であり，これだけから始端も同時に推定するのは困難である．以下の例では,「この辺りは父から聞いた話なんですけど」の部分だけを見た場合,「（他に自分が体験したことを話している途中で）この辺り（の話）は父から聞いた話なんですけど」という意味でも解釈できるため,「父から」が引用節の始端であるとは決定できない．この場合,「この辺りは父から聞いた話なんですけど」の全体が挿入節に含まれる可能性もある．

例）この┐
　　辺りは┐
　　（父から┐
　　聞いた┐
　　話なんですけど）
　　昔┘
　　たんぼだったんです

そこで本研究では，始端を決定する際には，自動付与した係り受けの情報をあわせて利用する．終端が分かっている場合，始端以前の係り受けついては，図 2 のような制約が成り立つ．よって，最初のチャンキングで終端を推定した後，以下の確率を素性に加え，再度チャンキングを行う．

(4) 自分より前の文節が，自分と終端の間に係る確率
(5) 自分の直前の文節が，終端より後方に係る確率

図 2 から，(4) の確率が小さく (5) の確率が大きければ，その文節は始端になりやすいといえる．上の例では,「辺りは」「聞いた」「話なんですけど」は手前の文節が自分に係るため，(4) の確率が大きくなる．また「父から」については，直前の文節「辺りは」が挿入節の終端「話なんですけど」より後方に係るため，(5) の確率が大きくなる．これより,「父から」が挿入節の始端であると推定できる．

4. 評 価 実 験

ここでは，引用節・挿入節認定および係り受け解析の評価実験を行った結果と考察について述べる．実験に用いたコーパスは CSJ188 講演の書き起こしである．このうち 168 講演を学習データ，20 講演をテストデータとして用いた．まず，下岡らの手法により文境界推定および係り受け解析を行い，ベースライン精度を求めた．文境界推定については F 値 85.9 で，推定された文ごとに係り受け解析を行った結果，open テストで，77.7 %, closed テストで 86.5 % の精度が得られた．

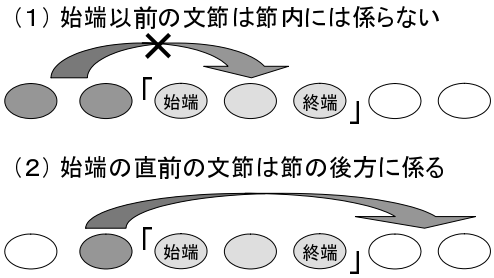

図 2　引用節・挿入節の始端以前の係り受け

(a) 引用節・挿入節認定結果

前章の手法を用いて，引用節・挿入節の自動認定を行った結果を表 2 に示す．表 2 には，係り受けを用いない場合 (1 回目のチャンキング)，open および closed で推定した係り受けを利用した場合 (2 回目のチャンキング) のそれぞれの結果を示した．また，1 回目のチャンキングで終端が正しく認定された割合も示した．

表 2 によると，引用節の終端はほとんど検出できている．検出できなかったものの中には「〜と」で終わる文末や,「〜っちゅう」「〜みたいな」など使われる頻度が比較的少ない表層表現があった．始端とともに正解した割合については，係り受けを利用することによって精度が向上した．これは，本手法で素性として利用した係り受けが有効に働いたことを表している．以下の例では，1 回目のチャンキングでは「自分のいい長所じゃないか」の部分が引用節だと誤って認定されたものの，2 回目のチャンキングで係り受けを利用することにより,「これは自分のいい長所じゃないか」の範囲が引用節であると正しく認定されるようになった．

例）{これは ┐
　　自分の┐
　　いい┐
　　長所じゃないか} と┐
　　私は┐
　　思います

しかし，利用した係り受けが open の場合と closed の場合とを比較すると引用節認定の F 値に 9.1 もの差がある．このことから，open の場合の係り受け解析では引用節に関わる誤りが多いことが分かる．このとき，1 回目のチャンキング結果が正しかったにもかかわらず,誤った係り受けを利用することで，2 回目のチャンキングで誤った結果となる場合もあった．

表 2　引用節・挿入節の認定精度 (文境界が未知の場合)

	引用節			挿入節		
	再現率	適合率	F 値	再現率	適合率	F 値
係り受けを利用しない	41.1 % (264/643)	44.3 % (264/596)	42.6	1.3 % (1/76)	20.0 % (1/5)	2.5
係り受けを利用 (open)	42.1 % (271/643)	45.5 % (271/596)	43.7	2.6 % (2/76)	40.0 % (2/5)	4.9
係り受けを利用 (closed)	50.9 % (327/643)	54.9 % (327/596)	52.8	2.6 % (2/76)	40.0 % (2/5)	4.9
終端のみ一致	89.1 % (573/643)	96.1 % (573/596)	92.5	2.6 % (2/76)	40.0 % (2/5)	4.9

表 3　係り受け解析精度 (文境界が未知の場合)

	open	closed
引用節・挿入節を利用しない	77.7 %	86.5 %
引用節・挿入節 (推定結果) を利用する	78.5 %	86.6 %

一方，挿入節については，係り受けを利用してもほとんど検出できていなかった．このとき，挿入節の終端の大半は文境界であると推定されていた．挿入節は「〜けれども」「〜ですが」の形で終わるものが多いが，これらの表現は文末にも用いられる．本手法で用いた素性だけでは，挿入節の終端と文末表現を区別できていない．

(b) 認定結果を用いた係り受け解析結果

続いて，open の係り受けを利用して推定された引用節・挿入節を用いて係り受け解析を行ったところ，表 3 のような結果となった．引用節・挿入節の推定精度が十分な値でないにもかかわらず，これらの情報を利用することで，open での係り受け解析精度が 0.8 % 向上した．これは，引用節・挿入節の推定に誤りがある場合でも，係り受け解析モデルが頑健に作用しているためと考えられる．以下の例では,「挟んで」(引用節内部) が「覚えてきて」(引用節外部) に係っていたものが，推定した引用節の情報を利用することで,「出てしまうという」(引用節内部) に係るように修正された．

```
例）{顔─┐
    挟んで──────────────────────┐
       外に─┐                   │
         出てしまう} という─┐    │
                     芸を──────┤
                      どこからか─┤
                          覚えてきて
```

(c) 文境界が既知の場合の実験結果

次に，文境界推定の誤りの影響を調べるため，正解の文境界を与えて，引用節・挿入節認定および係り受け解析を行った．結果は，表 4, 表 5 のようになった．ここで表 5 には，引用節・挿入節の正解を与えた場合の係り受け解析精度も示した．

文境界を与えたことにより，引用節・挿入節の推定精度，係り受け解析精度ともに大きく上昇した．また，引用節・挿入節の推定結果を用いることで，open での係り受け解析精度は 0.7 % 向上した．さらに，引用節・挿入節の正解を与えた場合，open で約 2 % , closed で約 1 % の向上に至った．

表 4　引用節・挿入節の認定精度 (文境界が既知の場合)

	引用節			挿入節		
	再現率	適合率	F 値	再現率	適合率	F 値
係り受けを利用しない	46.0 % (296/643)	50.8 % (296/583)	48.3	22.4 % (17/76)	23.6 % (17/72)	23.0
係り受けを利用 (open)	46.7 % (300/643)	53.3 % (300/563)	49.8	30.3 % (23/76)	38.3 % (23/60)	33.8
係り受けを利用 (closed)	55.1 % (354/643)	62.9 % (354/563)	58.7	30.3 % (23/76)	39.0 % (23/59)	34.1
終端のみ一致	86.5 % (556/643)	95.4 % (556/583)	90.7	64.5 % (49/76)	68.1 % (49/72)	66.2

表 5　係り受け解析精度 (文境界が既知の場合)

	open	closed
引用節・挿入節を利用しない	81.0 %	90.3 %
引用節・挿入節 (推定結果) を利用する	81.7 %	90.3 %
引用節・挿入節 (正解) を利用する	82.8 %	91.2 %

以上の結果から，引用節・挿入節は，文境界と同様に，話し言葉の係り受け解析精度に大きく影響する問題であり，これらの推定精度の改善が，係り受け解析精度の向上にもつながることが分かった．

5. おわりに

本稿では，CSJ を対象にして，引用節・挿入節の自動認定および認定結果を係り受け解析に適用する手法について述べた．実験により，引用節・挿入節の情報が係り受け解析精度を向上させることを確かめた．今後の課題としては，実験の考察を踏まえて精度の改善を図ることや，音声認識結果に誤りがある場合の頑健性について検討することなどが挙げられる．また，引用節・挿入節の情報を，文書整形などに応用することについても考えていきたい．

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) 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.
3) 下岡和也, 内元清貴, 河原達也, 井佐原均. 日本語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. 自然言語処理, Vol. 12, No. 3, pp. 3–18, 2005.
4) 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, Vol. 7, No. 5, pp. 3–17, 2000.
5) Taku Kudo and Yuji Matsumoto. Chunking with Support Vector Machines. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association of Computational Linguistics*, pp. 192–199, 2001.