統計的構文解析器を用いた音声言語係り受けコーパスの構築

大野 誠寛† 松原 茂樹‡ 河口 信夫‡ 稲垣 康善†

†名古屋大学大学院工学研究科情報工学専攻 ‡名古屋大学情報連携基盤センター/CIAIR

{ohno,matu,kawaguti,inagaki}@inagaki.nuie.nagoya-u.ac.jp

1 はじめに

係り受け情報が付けられた大規模テキストデータ (以下，係り受けコーパス) は，自然言語処理において重要な役割を果たしている．実際，新聞等の大規模言語データに基づく書き言葉の係り受けコーパス [3] は，構文解析はもちろん，情報検索，要約，機械翻訳など広く利用されている．それに比べ，話し言葉の係り受けコーパスは十分に整備されているとは言い難い．大規模な音声言語係り受けデータを作成し，それを統計情報として活用することにより，ロバストで精度の高い音声言語係り受け解析の実現が期待できる．しかしながら，係り受けコーパスの構築には，一般に，形態素分析，文節まとめあげ，係り受け構造の付与などといった作業が必要であり，それを人手で行うのは莫大な労力をともなう．

本論文では，統計的構文解析器を用いた音声言語係り受けコーパスの構築について述べる．本手法では，まず，形態素，及び，構文情報を付与したデータを自動的に作り上げ，それを人手により修正する．また，統計に基づく構文解析器は，学習データが増加するほど解析の精度が高くなることに着目し，本研究では，係り受け解析の結果に，人手で修正を加えることにより作成した係り受けデータを，別のテキストデータに構文情報を付与するための統計情報として利用するという増殖的なコーパス構築手法を採用した．構築対象のコーパスデータの全てに係り受け解析を施し，その後一括して人手で修正を行うよりも修正に要する労力の軽減が期待できる．

本手法に基づき音声言語係り受けコーパスを構築するために，ロバストな統計的係り受け解析手法を開発し [5]，名古屋大学 CIAIR 車内音声対話コーパスの 10,995 ターンのドライバー発話に対し，係り受け情報を付与する作業を実施した．

2 CIAIR 車内音声対話コーパス

名古屋大学 CIAIR では，1999 年度より車内音声対話の収録を実施している [2]．対話は，ドライバーとナビゲータとの間で遂行され，タスクとして店検索や道案内などが設定されている．収集した音声データの書き起こしは，日本語話し言葉コーパス (CSJ) [4] に準拠しており，作業は人手により行っている．書き起こしデータの例を図 1 に示す．データの言語学的分析として，フィラー，言い淀み，言い誤りなどにタグを付与している．さらに，各対話をポーズで分割し，各々を発話単位としてその開始時間，終了時間を記録している．また，各発話には，話者の性別 (男性/女性)，話者役割 (ドライバー/ナビゲータ)，対話タスク (道案内/情報検索など)，雑音状況 (有/無) に関する情報も付与されている．

```
0009 - 00:41:960-00:50:228 F:O:I:R:
この先                  &       コノサキ
三百メートルほど先に    &       サンビャクメートルホドサキニ
左手に                  &       ヒダリテニ
サンクス                &       サンクス
五百メートルほど先に    &       ゴヒャクメートルホドサキニ
セブンイレブンが        &       セブンイレブンガ
ございます<SB>          &       ゴザイマス<SB>
0010 - 00:51:888-00:52:100 F:D:P:C:
(F ん)                  &       (F ン)
0011 - 00:54:633-00:55:394 F:D:P:R:
(? 次左かな)            &       (? ツギヒダリカナ)
0012 - 01:02:750-01:03:993 F:O:I:R:
どちらに                &       ドチラニ
なさいますか<SB>        &       ナサイマスカ<SB>
```

図 1: 車内音声対話コーパスの書き起こしデータ

3 音声言語係り受けコーパス

CIAIR 車内音声対話コーパスのドライバーの発話に対して，以下の情報を付与することにより，音声言語係り受けコーパスを作成した．

- 形態素情報
  - 形態素区切り
  - 形態素の読み，原形，品詞，活用型，活用形
- 構文情報
  - 文節区切り
  - 文節間の係り受け

ここで，品詞体系は形態素解析器茶筌 [6] の IPA 品詞体系 [1] に，文節の区切りは日本語話し言葉コーパスの作成基準 [4] に，係り受け文法は京大コーパスの作成基準 [3] にそれぞれ準拠した．ただし，話し言葉特有の現象については，以下の作成基準を設けた．

- フィラーや言い淀みは，係り先が存在しない．すなわち，単独で係り受け構造を形成する．
- 受け文節が省略された文節の係り先は存在しない．
- 話し言葉特有の言い回し表現 (「こっから」,「っていうか」など) については，新たな辞書項目を設けて，形態素ごとに品詞を定める．
- 対話ターンを係り受けの単位とする．すなわち，発話単位を越える係り受けも認めるが，対話ターン間にまたがった係り受けは認めない．

```
((1 ((きょう キョウ きょう 名詞 副詞可能 なし なし)))
 -> (2 ((朝 アサ 朝 名詞 副詞可能 なし なし))))

((2 ((朝 アサ 朝 名詞 副詞可能 なし なし)))
 -> (4 ((食べ タベ 食べる 動詞 自立 一段 連用形)
        (て テ て 助詞 接続助詞 なし なし))))

((3 ((パン パン パン 名詞 一般 なし なし)))
 -> (4 ((食べ タベ 食べる 動詞 自立 一段 連用形)
        (て テ て 助詞 接続助詞 なし なし))))

((4 ((食べ タベ 食べる 動詞 自立 一段 連用形)
     (て テ て 助詞 接続助詞 なし なし)))
 -> (7 ((食べ タベ 食べる 動詞 自立 一段 連用形)
        (た タ た 助動詞 なし 特殊・タ 基本形)
        (ん ン ん 名詞 非自立 なし なし)
        (です デス です 助動詞 なし 特殊・デス 基本形)
        (よ ヨ よ 助詞 終助詞 なし なし))))

((5 ((お昼 オヒル お昼 名詞 副詞可能 なし なし)
     (は ハ は 助詞 係助詞 なし なし)))
 -> (6 ((お オ お 接頭詞 名詞接続 なし なし)
        (そば ソバ そば 名詞 一般 なし なし)
        (を ヲ を 助詞 格助詞 なし なし))))

((6 ((お オ お 接頭詞 名詞接続 なし なし)
     (そば ソバ そば 名詞 一般 なし なし)
     (を ヲ を 助詞 格助詞 なし なし)))
 -> (7 ((食べ タベ 食べる 動詞 自立 一段 連用形)
        (た タ た 助動詞 なし 特殊・タ 基本形)
        (ん ン ん 名詞 非自立 なし なし)
        (です デス です 助動詞 なし 特殊・デス 基本形)
        (よ ヨ よ 助詞 終助詞 なし なし))))

((7 ((食べ タベ 食べる 動詞 自立 一段 連用形)
     (た タ た 助動詞 なし 特殊・タ 基本形)
     (ん ン ん 名詞 非自立 なし なし)
     (です デス です 助動詞 なし 特殊・デス 基本形)
     (よ ヨ よ 助詞 終助詞 なし なし)))
 -> (NO (なし)))
```

図 2: 音声言語係り受けコーパス

係り受けコーパスの例を図 2 に示す．係り受け関係の列で表し，各係り受け関係は，係り文節と受け文節からなる．各文節には，文節番号とその文節を構成する形態素を列挙する．

4 係り受けコーパスの構築

本研究では，統計に基づく係り受け解析の結果を人手で修正することにより，大規模係り受けコーパスを構築する．効率的な構築作業を実現するために，修正の労力をできる限り抑えることが重要であるが，修正量の程度は，使用する係り受け解析の性能に大きく依存する．精度の高い係り受け解析を実行することが望まれ，一般にはそれは，統計情報を獲得するために用いる言語データの規模を増やすことによって実現可能となる．

そこで本稿では，係り受けデータ付与の対象となるコーパスの全てに対して一括して係り受け解析を行うのではなく，コーパスを複数に分割し，逐次的に構築作業を遂行することによって，増殖的な係り受けコーパスの作成を実現する．すなわち，係り受け解析の結果に人手で修正を加えて作成したデータを，すでに構築されている係り受けコーパスに追加し，別のテキストコーパスを係り受け解析するときの統計情報として利用する．統計情報は修正したデータから自動的に獲得できるので，解析結果の修正以外に人手による作業を必要としない．コーパス構築の過程で，係り受け解析の精度が漸増的に向上し，コーパス修正に要する労力を軽減することが可能となる．

4.1 コーパス構築の流れ

音声言語係り受けコーパス構築の流れを図 3 に示す．複数セットに分割された CIAIR 車内音声対話コーパスから 1 つを取り出し，以下の作業を順に行う．

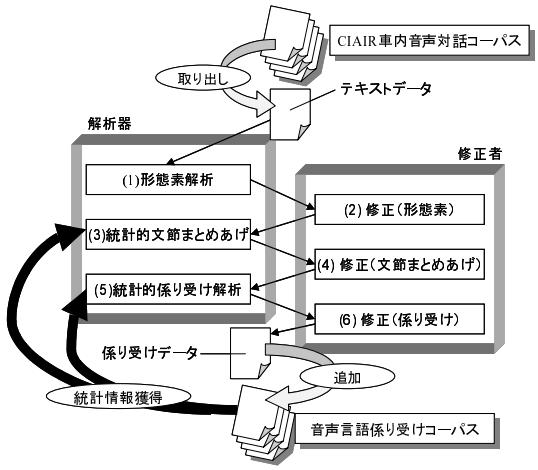

図 3: 音声言語係り受けコーパス構築の流れ

(1) 音声データの書き起こしテキストを，形態素解析システム茶筌 [6] により形態素に区切り，形態素分析を与える．

(2) 形態素情報を人手により修正する．話し言葉特有の言い回しや固有名詞が新たに出現した場合は，茶筌の辞書に随時追加する．

(3) 統計情報を利用した文節まとめあげ手法を用いて，文節ごとに区切る．

(4) 文節区切りの修正を人手により行う．

(5) 統計的な係り受け解析により，係り受け構造を付与する．

(6) 係り受け情報を人手により修正を加える．

この (6) の結果を音声言語係り受けコーパスに追加する．

以下では，本構築手法で用いた統計的な文節まとめあげ手法，及び，係り受け解析手法について述べる．

4.2 統計的文節まとめあげ

文節まとめあげとは，形態素列 $M(= m_1 \ldots m_n)$ の各形態素間に文節区切りを挿入するか否かを判定する問題である．例えば，「今日朝パン食べてお昼はおそばを食べたんですよ」に対する文節まとめあげの効果は次の通りである．

今日/朝/パン/食べ␣て/
お昼␣は/お␣そば␣を/食べ␣た␣ん␣です␣よ

本手法では，文節まとめあげ済みコーパスから，文節区切りに関する統計情報を獲得し，それを用いて文の先頭から順に判定することを試みる．隣接する形態素 m_i と m_{i+1} の間の文節区切りに注目するとき，統計情報として各形態素の以下に挙げる属性を利用する．

- 形態素の原形 h_i, h_{i+1}
- 形態素の品詞 t_i, t_{i+1}
- 形態素の活用形または品詞細分類 s_i, s_{i+1}

形態素 m_i と m_{i+1} の境界で文節区切りとなる確率，すなわち，同一文節に含まれない確率を以下のように計算する．

$$P(m_i/m_{i+1}|m_i, m_{i+1}) \quad (1)$$
$$= \frac{C(m_i/m_{i+1}, h_i, h_{i+1}, t_i, t_{i+1}, s_i, s_{i+1})}{C(h_i, h_{i+1}, t_i, t_{i+1}, s_i, s_{i+1})}$$

ここで，m_i/m_{i+1} は，連続する形態素 m_i, m_{i+1} の間に文節の区切りが存在することを意味する．C は共起頻度関数である．ここでは，$P(m_i/m_{i+1}|m_i, m_{i+1}) \geq 0.5$ ならば，そこで文節を区切る．

4.3 統計的係り受け解析

著者らが提案する統計的な係り受け解析手法 [5] では，倒置や受け文節のない係り受けなどの話し言葉に特有な現象もロバストに解析できる [7]．この手法では，文節列 B $(= b_1 \cdots b_n)$ の係り受け構造を S とするとき，$P(S|B)$ の確率値を最大にする係り受け構造 S を求める．

通常の書き言葉に対する係り受け解析手法では，係り受けの非交差性，後方修飾性，係り先の唯一性の 3 つの性質を絶対的制約として用いる．本手法では，倒置やフィラー，言い淀み，言い誤りなどが頻出することを考慮して，非交差性のみを満たすべき性質として，係り受け構造を求める．ただし，後方修飾性，及び，係り先の唯一性の充足については，統計情報を反映することにより考慮する．

それぞれの係り受けは独立であると仮定すると，$P(S|B)$ は以下の式で計算できる．

$$P(S|B) = \prod_{i=1}^{n} P(b_i \overset{rel}{\rightarrow} b_j|B) \quad (2)$$

ここで，$P(b_i \overset{rel}{\rightarrow} b_j|B)$ は，入力文節列 B が与えられたときに，文節 b_i から b_j への係り受け関係が存在する確率である．最尤の係り受け構造は，式 (2) の確率を最大とする構造であるとして動的計画法を用いて計算する．

次に，$P(b_i \overset{rel}{\rightarrow} b_j|B)$ の計算について述べる．まず，係り文節における自立語の原形を h_i，その品詞を t_i，係りの種類を r_i とし，受け文節における自立語の原形を h_j，その品詞を t_j とする．また，文節間距離を d_{ij}，文節間のポーズの数を p_{ij}，係り文節の位置を l_i とする．ここで，係りの種類とは，係り文節が付属語を伴うときはその付属語の語彙，品詞，活用形であり，そうでない場合は一番最後の形態素の品詞，活用形である．また，係り文節の位置は，その文節が入力ターン内で一番最後の文節か否かを表す．

以上の属性を用いて，確率 $P(b_i \overset{rel}{\rightarrow} b_j|B)$ を以下のように計算する．

$$P(b_i \overset{rel}{\rightarrow} b_j|B) = \frac{C(b_i \rightarrow b_j, h_i, h_j, t_i, t_j, r_i, d_{ij}, p_{ij}, l_i)}{C(h_i, h_j, t_i, t_j, r_i, d_{ij}, p_{ij}, l_i)} \quad (3)$$

ただし，C は共起頻度関数である．係り先のない文節はそれ自身に係る (すなわち $i = j$) とみなすことにより，係り先をもたない場合の確率も計算できる．

5 解析結果修正用インタフェース

人手による修正の負担を軽減するため，GUI ベースの修正インタフェースを作成した．インタフェース画面

図 4: 解析結果修正用インタフェース

を図 4 に示す．これを用いて全ての修正を行うことができる．

インタフェースの左側の画面で形態素情報と文節まとめあげの解析結果を修正する．1 つの形態素が 1 行で表現され，各行には，形態素の各種情報を表示したボタンが並んでいる．品詞，品詞細分類，活用型，活用形はそれぞれメニューボタンで表示され，メニューバーから適切なものを選択することにより修正できる．

また，同一の文節に含まれる形態素をすべて同一の色で示すことにより，文節のまとまりを表現する．文節区切りの修正は，画面の左端にある操作ボタンをクリックし，その行の色を変更することにより行う．変更後に係り受け修正ボタンを押すと，係り受け解析プログラムが動作し，修正内容が係り受け構造に反映される．

インタフェースの右側の画面で係り受け構造を修正する．この画面では，文節とその係り先文節を示すとともに，係り受け関係を視覚的に表示している．係り先の文節番号が「NO」となっている場合は，その文節が係り先をもたないことを意味する．

6 評価実験

係り受け関係を自動的に与えたときのコーパス全体の修正に要する労力に着目し，本稿で提案した構築手法の評価を行った．

6.1 実験の概要

実験用データとして，CIAIR 車内音声対話コーパスの 221 対話を使用した．データの規模は，45,053 文節からなる 10,995 ターンである (平均ターン長は 4.1 文節)．このうち，1,074 ターンを基礎学習データとし，残りの 9,921 ターンをテストデータとした．増殖的構築の効果を調べるために，テストデータを 10 個のセット (1 セットは約 1,000 ターン) に等分割した．便宜上，各セットに 1 から 10 までの番号を与えた．

同一のデータに対し，以下の 2 つの手法でコーパスを構築した．

- **増殖的構築手法**　4 節で述べた手法に基づいて係り受けデータを付与した．セット 1 から 10 まで順に構築作業を実施した．各セットへの係り受け付与

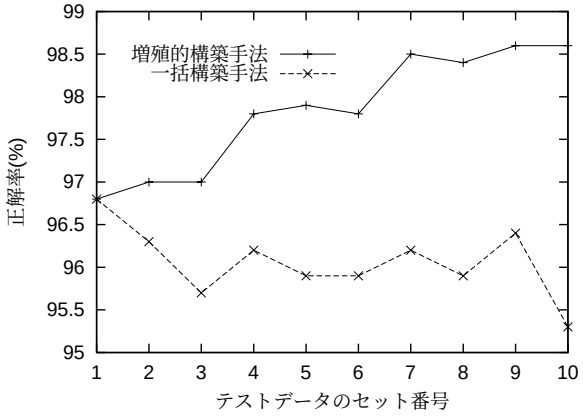

図 5: 統計的文節まとめあげの実験結果

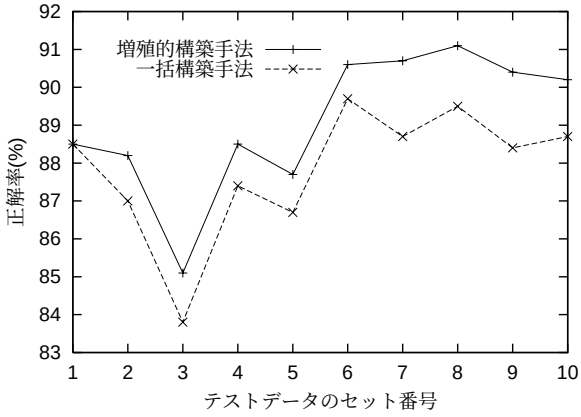

図 6: 統計的係り受け解析の実験結果

が完了したデータは，それ以降の解析の学習データに追加した．

- **一括構築手法**　全てのテストデータに対し，基礎学習データのみを用いて係り受け解析を行い，その後，一括して人手により修正した．ただし，文節まとめあげと係り受け解析は 4 節で述べた手法と同一である．

評価は，文節まとめあげと係り受け解析の精度を両構築手法の間で比較することにより行った．

6.2　実験の結果

両手法による文節まとめあげの結果を図 5 に示す．増殖的構築手法の方が 1.72%高い精度でまとめあげを行うことができた．テストデータに含まれる形態素間の境界は計 87,712 個所あり，そのうち正しく判定できた数は，一括構築手法では 84,243 個所，増殖的構築手法では 85,751 個であった．すなわち，増殖的構築手法を採用することにより，修正個所が 1,508 個削減されており，これは，修正量が一括構築手法の 6 割程度で済むことを意味している．

係り受け解析結果の比較を図 6 に示す．増殖的構築手法の正解率は 89.0%，また，一括構築手法の正解率は 87.8%であり，文節まとめあげと同様，増殖的構築手法のほうが高い精度で係り受けデータの付与が行えた．テストデータに含まれる係り受け数 40,776 個のうち，正しくデータを付与できた数は，一括構築手法が 35,787 個で，増殖的構築手法は 36,300 個であった．すなわち，増殖的構築手法を採用することにより，修正個所が 513 個削減されており，これは，一括構築手法と比較し，修正量を約 1 割削減できたことを意味する．

これらのことから，コーパス全体の修正に要する労力の軽減に本稿で提案した増殖的構築手法が有効であることが確認できた．

7　おわりに

本稿では，統計的構文解析器を用いた音声言語係り受けコーパスの増殖的な構築について述べた．本手法では，係り受け解析の結果に，人手で修正を加えることにより作成した係り受けデータを，別のテキストデータに構文情報を付与するための統計情報として利用する．評価の結果，コーパス構築時における人手修正に要する労力の軽減に，本手法が有効であることを確認した．

本手法の効果は，コーパスの規模が増すほど大きく表われ，修正に要する人的負担は次第に軽減すると予想される．今後も引き続き係り受けコーパスの構築を進めるとともに，データを用いた詳細な分析については稿を改めて報告する予定である．

謝辞　係り受けコーパスの修正を御協力いただいた本学大学院国際言語文化研究科の遠山仁美さんに感謝致します．本研究の一部は，文部省科学研究費補助金 COE 形成基礎研究費 (課題番号 11CE2005，代表 名古屋大学大学院工学研究科 板倉文忠教授) の補助を受けて行われた．

参考文献

[1] 浅原 正幸，松本 裕治: IPADIC ユーザーズマニュアル, version2.5.1 (2002).

[2] Kawaguchi, N., Matsubara, S., Takeda, K. and Itakura, F.: Multimedia Data Collection of In-Car Speech Communication, *Proc. of Eurospeech-2001*, pp.2027–2030 (2001).

[3] 黒橋 禎夫, 長尾 真: 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).

[4] 前川 喜久雄, 籠宮 隆之, 小磯 花絵, 小椋 秀樹, 菊池 英明: 日本語話し言葉コーパスの設計, 音声研究, Vol.4, No.2, pp.51-61 (2000).

[5] Matsubara, S., Murase, T., Kawaguchi, N. and Inagaki, Y.: Stochastic Dependency Parsing of Spontaneous Japanese Spoken Language, *Proc. of COLING-2002*, Vol.1, pp.640-645 (2002).

[6] 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』, version2.2.9, 使用説明書 (2002).

[7] 大野 誠寛，松原 茂樹，河口 信夫，稲垣 康善: 日本語音声対話文の統計的係り受け解析とその評価, 情報処理学会第 65 回全国大会講演論文集 (2003).