分類木を用いた日本語長文の自動分割

張 玉潔　尾関 和彦

電気通信大学

{zhang, ozeki}@achilleus.cs.uec.ac.jp

1 はじめに

多数の接続節を含む日本語長文を，そのまま構文解析することは非常に困難である．そのため，構文解析の前処理として，長文を短文に分割することが試みられている[1]．また，長文は人にとっても理解しにくい悪文であることが多いため，推敲支援の観点からも短文分割の研究がなされている[2]．これら従来手法の要点は，次のようにまとめることができる．

(1) 形態素解析により得られる品詞や表記情報を用いて，分割点を推定するための分割パターンを表現する．そして，入力文と分割パターンとのマッチングにより，分割点を推定する[1]．

(2) 従属句の間の包含関係[3]により述語間の係り受け構造の候補を挙げ，さらに述語間の係り易さを表すヒューリスティックなスコアを用いて，それらに順位を付ける．そして，順位の高い構造を選ぶことにより，分割点を推定する[2]．

これらの手法は，それぞれに有効であることが報告されているが，分割パターンや述語間の係り易さを表すスコアなどを人手で作成・設定しなければならないという問題がある．

本論文では，形態素解析から得られる表層情報に基づき，分類木[4],[5],[6]の手法を用いて，分割パターンをコーパスから自動的に獲得する方法を提案する．この方法によれば，学習データ中に現われる言語現象とその出現頻度に応じて，最適分割パターンとその適用順序が自動的に決定される．以下では，この方法について説明し，EDRコーパスを用いた実験結果について報告する．

2 分類木

ここで用いた分類木は，大略次のようなものである．

(1) 中間節点がすべて2つの子節点を持つ2分木である．

(2) 不純度の計量には *Gini* Index[4]を採用している．

(3) 生成する際，ある節点において，不純度を減少させるようなテストが存在しなければ，その節点を葉とする．

(4) 葉には，多数決により「分割点である」か「分割点でない」かのラベルを付ける．

データとして何を考えるか，また，各節点でどのようなテストを行なうかについては，次節以降で詳しく説明する．

3 データ

3.1 分割点

長文分割問題を考えるとき，まず，正しい分割点の定義を明確にしておく必要がある．ここでは用言を含み文末の文節に係る文節と直後の文節の間の境界を分割点と定義する．文をこのような点で分割することによって得られるそれぞれのセグメントは，相互に独立した接続節である．

3.2 要素文節

文中の文節には，分割点を推定する上で重要な働きをするものと，あまり重要な働きをしないものがある．末尾が係助詞「は」,「も」，格助詞「が」である文節，および用言を含む文節は重要な働きをすると考えられるので，これらを文から抜き出し，それらの属性を抽出する．属性としては，

(1) 接続，(2) スコープ，(3) 読点

を考える．接続属性の属性値は，文節の主辞と末尾の形態により，表1のように定める．

表1 接続属性

属性値	主辞	文節末
動連用	動詞, 体言判定	連用形
て連用	用言	「て」,「で」
ため	用言	形式名詞, 時名詞
A	用言	接続助詞「ながら」など
B	用言	接続助詞「ば」など
C	用言	接続助詞「が」など
形動連用	形容動詞	連用形
形連用	形容詞	連用形
動連体	用言	連体形
用言	用言	助詞
形連体	形容詞, 形容動詞	連体形
は	体言	助詞「は」
も	体言	助詞「も」
が	体言	助詞「が」
終止文節	用言	「.」

属性値欄の「A」,「B」,「C」は文献[3]における接続形式の分類に対応している．主辞欄の「体言判定」は，体言に判定詞が付属したものを表す．また,「用言」は「動詞」,「形容詞」,「形容動詞」,「体言判定」を含んでいる．スコープ属性の属性値は，引用助詞「と」や形式名詞「こと」などが含まれるとき「スコープ」，含まれないとき「$NULL$」と定める．また，読点属性の属性値は，文節直後に読点「,」がある場合には「読点」，ない場合には「$NULL$」と定める．上記のような文節をこのような属性値の組で表したものを，ここでは要素文節と呼ぶ．文を要素文節の列に変換した例を示す．要素文節の添字は文中の文節番号を示す．

[原文]

１６日に 米 船籍 タンカーが 被弾した 時, クウェート軍は ミサイルの 飛来を 探知, 自軍の 地対空ミサイルで 迎撃しようとしたが, 失敗に 終わった.

[要素文節列]

$(が, NULL, NULL)_4$ $(動連体, NULL, NULL)_5$ $(ため, NULL, 読点)_6$ $(は, NULL, NULL)_7$ $(動連用, NULL, 読点)_{10}$ $(C, スコープ, 読点)_{13}$ $(終止文節, NULL, NULL)_{15}$

3.3 分割点候補

要素文節の中で，接続属性の値が「動連用」,「て」,「ため」,「A」,「B」,「C」,「形動連用」,「形連用」のいずれかであるものは，その直後が分割点になる可能性がある．そこで，これらの文節を分割文節とし，直後の文節との境界を分割点候補とする．ただし,直後の文節が文末の文節である場合には，推定する必要がないので,分割点候補としない．

3.4 データ

分割点を推定する上で，分割文節の属性値が重要であることは明らかである．また，分割点候補が分割点になるか否かは，分割文節がどの文節に係るかによって決まる．したがって，分割文節より後に現われる要素文節も重要である．しかし，分割文節より前に現われる文節は，あまり重要でないと考えられる．そこで，本研究では，分割文節以降の要素文節のみをテストの対象とする．分類木の入力となるデータは，分割文節以降の要素文節列と一つの分割点候補の組である．したがって，n 個の分割文節を持つ要素文節列からは，n 個のデータが作られる．学習データと評価データには，分割点候補が正しい分割点であるか否かによって，'YES', 'NO' のラベルが付けられる．

4 テスト

まず，テスト項目の集合 T を (接続属性値の集合 $\cup \{*\}) \times \{$ スコープ$, NULL, *\} \times \{$ 読点$, NULL, *\}$ と定義する．ここで，'*' は属性値が「何でもよい」ことを表す．また,「長さ１以上の任意の要素文節列」を表す記号'+'を導入する．そうすると，一つのテストは $[X] < Y >$という形で表される．X は $T \cup \{+\}$ の要素，Y は $T \cup \{+\}$ の要素の列で，'+' が連続しないものである．$[X]$ は分割文節がパターン X にマッチするかどうかをテストすることを表し，また，$< Y >$ は分割点候補より後の要素文節列がパターン Y にマッチするかどうかをテストすることを表わす．例えば，テスト

$[(A, *,$ 読点$)]$ $<$ (て連用$, *, *)$ + ($*$, スコープ$, *)+ >$

に合格するデータは，分割文節の接続属性が「A」であって「読点」属性を持ち，かつ，分割点候補直後の要素文節の接続属性が「て連用」であり，かつ，分割点候補直後の要素文節と最後の要素文

節の中間に「スコープ」属性を持つ要素文節が存在するようなものである．'$*$' の部分の属性値は問われない．

各節点におけるテストは，分類木を生成する過程で，次のように定められる．ある節点に到達する学習データがそれ以前の段階で既に合格しているテストを，その節点の既知テストと呼ぶ．既知テスト $[X] < Y >$ における $[X]$ の中の '+' を 't' ($t \in T$) に，また，$< Y >$ の中の '+' を 't', '$+t$', '$t+$', '$+t+$' ($t \in T$) に順次置き換えることにより，新たなテストを生成する．これらの生成されたテストの中から不純度減少規準[4]により最適なものを選択し，この節点のテストとする．この節点を，テストに合格するデータが集まる 'yes' 子節点と，テストに合格しないデータが集まる 'no' 子節点に展開する．'yes' 子節点の既知テストは親のテストに等しく設定し，'no' 子節点の既知テストは親の既知テストに等しく設定する．根の既知テストを $[+] < + >$ に初期設定し，上のような手続きを停止条件が満たされるまで再帰的に繰り返すことにより，分類木が生成され，各節点のテストが決定される．

5 実験

EDR コーパス[7]に対し，上に述べた方法によって，分類木の生成と分割点推定実験を行なった．

5.1 データの作成

コーパスから長さが30形態素以上の2000文をランダムに選んで，括弧で示された各文の構文情報から係り受け構造を抽出した．その結果，唯一性と非交差性を満す1847文が得られた．同様の構文情報を用いて文節区切りを行ない，各文節の主辞を定めた．また，文節末の活用語に対して，付属単語辞書の活用情報により活用形を抽出した．これらの結果を用いて，1847文を要素文節列に変換し，分割点候補を検出し，データを作成した．さらに，もう一度構文情報を用いて，分割点候補が正しい分割点であるか否かによって，各データに 'YES', 'NO' のラベルを付与した．総データ数は2316個となった．

5.2 分類木の生成と分割点推定実験

1847文のうちランダムに選んだ400文の651個のデータを評価データとし，残りの1447文の1665個を学習データとした．生成された分類木の節点数は611個で，葉は306個になった．根に近い部分を図1に示す．

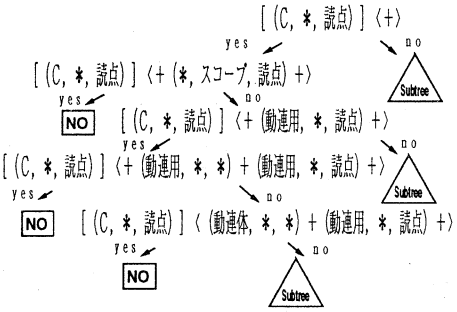

図1 分類木の根に近い「C」接続属性に関する部分

この分類木により，例えば，分割文節が接続属性「C」と読点属性「読点」を持つデータは，'yes' 子節点に振り分けられる．そこで，分割点候補の後の要素文節列がチェックされ，もし直後の要素文節と文末の要素文節の中間に「スコープ」と「読点」を持つ要素文節があれば，'NO' というクラスラベルを持った葉に到達する．これにより，そのデータの分割点候補は分割点ではないと推定される．

評価データを用いて，この分類木による分割精度を評価した．400文の中に分割点候補がない文は7文あった．分割結果の例を以下に示す．'?' は分割点候補を，その後の数字は分割点候補番号を，(Y) と (N) は「分割点である」か「分割点でない」かの推定結果を，また，'|' は正しい分割点を表す．

[例文1] １６日に 米 船籍 タンカーが 被弾した 時, $?_1(N)$ クウェート軍は ミサイルの 飛来を 探知, $?_2(N)$ 自軍の 地対空ミサイルで 迎撃しようとしたが, $?_3(Y)$ | 失敗に 終わった．

[例文2] 病が 進み, $?_1(Y)$ スタジオの ソファに 横に なりながら $?_2(N)$ 指示を 出していた 亀井さんは，最後の ロールが 終わったとき，涙ぐんだ．

[例文 3]休みを 取らなければ,$?_1(N)$ 次々に 加算されていくので, $?_2(Y)$ まとめて $?_3(N)$ 木, 金曜を 休みに して,$?_4(Y)$ 週休と 合わせて $?_5(N)$ 4連休に することも できる.

例文 1 では, すべての分割点候補における推定結果がラベルと一致したが, 例文 2 の '$?_1$', 例文 3 の '$?_2$', '$?_4$' のところは誤って分割点と推定された. しかし, 例文 3 において, もし文末文節が「できる」ではなくて,「することもできる」ならば, そこの二つの誤りは正しい推定結果と考えられる. EDR コーパスには,「でき」を付属語と考えてラベル付けしている文もあるので, 例文 3 における誤りは許容できる誤りである. 本来はラベルを修正すべきであるが, ここでは推定結果だけを人手で評価し直した.

以下の尺度により評価した結果を表 2 に示す. 括弧中の数字は人手で評価し直した結果である.

$$\text{適合率} = \frac{\text{正しく推定された分割点数}}{\text{推定された分割点数}}$$

$$\text{再現率} = \frac{\text{正しく推定された分割点数}}{\text{正しい分割点数}}$$

$$\text{文正解率} = \frac{\text{完全に正しく分割された文数}}{\text{評価文数}}$$

誤分割された例を調査したところ, 次のような問題があることが分かった.

(a) 分割点候補より後に接続属性「動連体」, あるいはスコープ属性「スコープ」 を持つ要素文節が存在する場合は誤りが多かった. これは, 連体節や引用節の範囲を推定するのが難しいことを示している.

表 2 評価実験の結果

評価文数	400
正しい分割点数	352
推定された分割点数	363
正しく推定された分割点数	294(343)
適合率 %	81 (95)
再現率 %	84 (97)
文正解率 %	72 (83)

(b) 情報の抽出処理が不十分である. 例えば, EDR コーパスでは助詞の分類が粗く,「と」 が, 引用, 接続, 並列のいずれの働きをするかの識別が十分でない.

(c) EDR コーパスの構文情報の不統一によって, 作成したデータが誤りとなった例がある.

6 おわりに

表層情報に基づく長文分割問題に対し, 分類木の手法を試みた. 提案した手法により, コーパスから分割点に関する表層情報のパターンを, 統計的な側面と論理的な側面から同時に捉えることができることが分かった.

今後は, より有効な表層情報の利用や, 分類木の適切な枝刈りによる汎化能力の向上などについて検討する予定である.

参考文献

[1] 金淵培, 江原輝将, "日英機械翻訳のための日本語長文自動短文分割と主語の補完," 情報処理学会論文誌, vol.35, no.6, pp.1018-1028, June 1994.

[2] 武石英二, 林良彦, "接続構造解析に基づく日本語複文の分割," 情報処理学会論文誌, vol.33, no.5, pp.652-663, May 1992.

[3] 南不二男, "現代日本語の構造," 大修館書店, 1974.

[4] L. Breiman et al., "Classification and Regression Trees," Chapman & Hall.

[5] 張玉潔, 尾関和彦, "分類木を用いた日本語文の自動文節分割" 情報処理学会研究報告, vol. 97, no. 85, pp.1-8, Sept. 1997.

[6] R. Kuhn and R. De Mori, "The Application of Semantic Classification Trees to Natural Language Understanding," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp.449-460, May 1995.

[7] 日本電子化辞書研究所, "EDR 電子化辞書 1.5 版仕様説明書," 1996.