係り受け関係コストを用いた高速な解探索手法に関する研究

福田憲司†　　森田和宏†　　泓田正雄†　　青江順一†

†徳島大学大学院　先端技術科学教育部

概要 本稿では，係り受け関係の強度をコストとし，一文全体でコストが最小となる解を高速に選択する係り受け解析手法について述べる．本手法では，文頭文節から各係り受け関係候補に対して，部分経路のコストを決定する．次に，A*アルゴリズムを用いて文末文節から，非交差条件，一文一格の制約などの文法規則を適用しながら部分経路を展開し，文全体のコストを求める．コストが最小となる解候補から順に展開することで，すべての解候補を計算するのに比べ高速である．本手法を用いた実験において，従来の係り受け解析器である KNP,Cabocha と比較して，1.4 倍から 1.8 倍の速度で解析できることを実証した．

1. はじめに

コンピュータの普及とインターネットの発達により，電子化された文書が増加している．これらの電子文書に含まれる情報を有効に利用する上で，文書を解析処理することが重要となる．構文解析は，形態素解析と同様に自然言語処理において，欠かすことのできない技術であり，基盤技術の一つと認識されている．

日本語の構文解析においては，係り受け解析を呼ばれる解析手法が用いられており，広く研究がおこなわれている．既存の日本語係り受け解析は，2 文節間の係り受け関係ルールを構築する手法[1]と，大規模なコーパスから 2 文節間の係り関係を機械学習して求める手法[2]がある．係り受け解析には，精度とともに，リアルタイム処理などで用いるために，解析速度も求められている．しかし，一文に複数現れる係り受け関係候補から最適な解を，効率的に探索する手法は見受けられない．

そこで，本稿では，各文節の係り先候補に係り受け関係の強度を表す重みとして，2 文節間にコストを設定し，このコストを用いて得られる複数の係り受け関係候補からコストが最小となる解を係り受け関係における制約条件を適用しながら効率良く探索する手法について述べる．これは，入力文に対する係り受け関係候補を尤もらしい順に N 個（N-best）求めることができる手法であり，先頭文節からの前向き部分経路のコスト計算と，末尾文節からの後ろ向き A*アルゴリズム探索により構成される．

2. 係り受け解析に用いる文節属性と制約

本手法では，係り受け関係を決定するために，入力文の文節に対して，いくつかの属性を決定する．そして，決定した属性を用いて，任意の 2 文節間にコストを付与し，係り受け関係に制約を適用することで，解の判定をおこなう．

2.1 文節に付与する属性情報

入力文を形態素および文節に分割した後，各文節に係り受け解析に用いる属性情報を付与する．各文節には，4 つの属性情報を付与し，係り受け解析に用いた．付与する属性情報には，まず文節の表記，文節に含まれる品詞情報，自立語の概念がある．そして，格助詞「の」句や，副助詞「は」句などの，文節の働きを表層格情報により定義している分類である句情報を用いる．

2.2 係り受け関係条件

係り受け解析では，以下に示す優先規則と日本語文法から考えられる係り受け関係における制約が知られている．本手法では，これらの関係を考慮に入れた解析をおこなう．

① 文節は係り得る最も近い文節に係る．
② 任意の文節は，その文節よりも後方にただ 1 つの係り先を持つ．
③ 同一文内の係り受け関係は互いに交差することはない．（非交差条件）
④ 同じ格要素が一つの用言に係ることはない．（一文一格の制約）

3. 係り受け関係コストと解探索手法

本解析手法では，先に述べたように，各文節に係り受け関係決定に必要となる属性情報を付与する．そして，任意の 2 文節間の係り受け関係を記述した

判定ルールと照合をおこない，係り受け関係の優先順位を決定するためのコスト付けをおこなう．付与したコストを用いて，尤もらしい順に序列化した解を決定する．

3.1 係り受け関係コスト決定法

係り受け関係コストは，前章で述べた文節属性情報を用いるコストと，格フレームにより求める 2 つのコストにより決定する．

文節属性情報に基づくコストは，任意の 2 文節間の関係を，先に決定した表記，自立語概念，句情報などの属性情報を基に，係り受け関係の強度をコストとして付与する．例えば，“<副詞句><動詞>”などのパターンを用いて，2 文節間の関係にコストを付与する．その際，文節間距離に応じて，距離が遠くなるほどコストを大きくする．この考えは優先規則①に基づいている．

格フレームに基づくコストは，任意の 2 文節間が格フレームに一致すれば，その 2 文節間の係り受けコストを低いコストとする．格フレームは，ある語とその語の取り得る格の制約を表しているので，係り受け関係が強いと考えられるからである．格フレームの作成に用いた体言概念は，日本語語彙体系に基づいて約 3,000 に体系付けられている[3]．

これら，文節属性情報と格フレームを用いた 2 文節間の関係とコストの 2 つを設定した係り受け辞書を作成し，解析に用いる．

3.2 解探索手法

形態素解析において，永田[4]は解析候補を最も尤もらしい順に任意の N 個（N-best）を求めることができる手法を提案した．これは，文頭から全ての部分経路のスコアを記録する前向きの動的計画法と，部分経路を展開する後ろ向き A*探索から構築した探索法である．

本手法は，形態素解析に用いられていた手法を係り受け解析に応用し，係り受け条件の制約を用いて解を探索する．解の探索は，先頭文節から始める前向きの動的計画法と，末尾文節から始める後ろ向き A*アルゴリズムにより構成される．

まず，前節で決定したコストを基にして，各文節の係り受け関係候補に対して連続する 2 文節間の部分経路のコストを計算する．次に，末尾文節から後ろ向きに部分経路を展開する．後ろ向き探索の部分経路コストと，前向き探索で求めた残りの経路に対する最小コストの和により，完全な経路のコストを算出することができる．また，コストが小さい解候補の部分経路から順に展開し，係り受け関係条件を考慮に入れることで，コストの最小性だけでなく係り受け条件の判定を効率良くおこなうことができる．

ここで，図 1 に解探索に用いる構造を示す．seg 構造とは，係り先の文節位置とコストを格納した構造である．この構造同士を結ぶことで解となる経路を作成する．順位テーブルとは，合計コストによって順位付けられる構造で，後ろ向き探索において最尤探索をおこなうために用いる．

3.2.1 前向き部分経路作成

前向き部分経路作成の例を図 2 に示す．前向きに合計コストを探索する部分経路の作成は，入力文の先頭文節から始まり，一文節ずつ文末に向かって進む．文節の各位置において，係り先候補文節のコストと，前文節の係り先候補文節とのコスト和を計算する．ここで，文節数を n とし，任意の文節 $i(0 < i \leq n)$ における係り先候補数を k_i とする．このときの係り先となる seg 構造を $S_{ij}(0 < i \leq n, 0 < j \leq k_i)$ で表す．S_{ij} の係り受けコストを C_{ij} とすると，任意の文節における最小コスト(L_{ij})は，その前文節の最小コストを用いて，以下の式(1)

seg 構造

係り先文節
係り受けコスト
合計最小コスト
その他合計コスト
:

順位テーブル（n：文節数）

係り文節	n	n-1	･･	1	合計コスト
受け候補文節 x	-	-	-	-	-
:	:	:	:	:	:

図 1：解探索に用いる構造

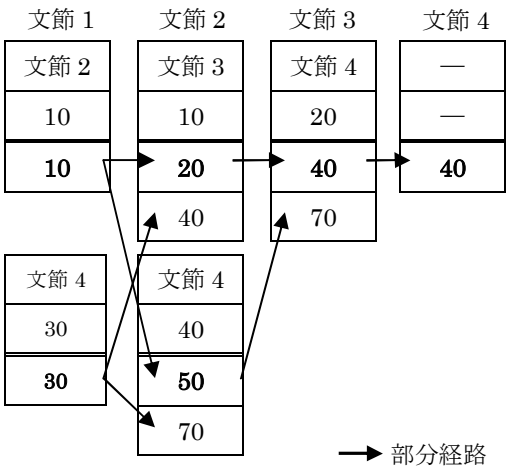


図 2：各文節の係り先候補とコスト

で表すことができる．min は範囲内の最小値とし，$L_{01} = 0$ とする．

$$L_{ij} = \min(L_{i-1m} + C_{ij} (0 < i \leq n, 0 < m \leq k_{i-1})) \quad (1)$$

上記の式(1)により，任意文節の係り先 seg 構造までの部分経路の最小コストが求められる．ただし，式(1)における最小コスト以外のコストについても経路を作成し，前文節の係り先 seg 構造の位置を格納しておく．つまり，任意の文節において，先頭からその文節までの最小コストを用いて，次文節との合計コストを算出し，経路を作成する．任意文節の係り先 seg 構造に対して，正確に最小コストが求められているので，末尾文節までの経路が作成できたとき，一文の最小コストが求められる．

3.2.2 後ろ向き解探索

後ろ向き解探索では，前項で述べた前向きに作成した部分経路構造と，合計コストによる順位テーブルを用いる．

先に決定した 2 文節間の係り先候補同士の部分経路を，文末から文頭に向けて展開する．任意文節において，その係り先候補のコストが小さい経路から順に展開し，文全体のコストを決定する．

任意文節の seg 構造における部分経路コストの合計は，その後ろ（文末側）すべての文節の任意の経路をただ 1 つだけ辿っている．よって展開中の経路のコスト C_{ij} の，そこまでの合計（M_{ij}）は，以下の式(2)により与えられる．ここで，末尾文節におけるコスト合計は，すでに求められている最小コストとなるため $M_{n1} = L_{n1}$ で与えられる．

$$M_{ij} = M_{i+1m} + C_{ij} \quad (0 < i < n, 0 < m \leq k_{i+1}) \quad (2)$$

ここで，すでに前向き探索により与えられた合計最小コスト L_{ij} との和により，残りの経路を含めた最小コスト（Sum_{ij}）を以下の式(3)により求めることができる．

$$Sum_{ij} = M_{ij} + L_{ij} \quad (3)$$

前向き探索により任意文節の seg 構造における最小コストを求めているので，Sum_{ij} より小さなコストの解が現れることはない．

図 2 に示した例について，コスト最小の解を探索した部分経路の展開結果と，このときの順位テーブルの内容を図 3 に示す．解探索手順を以下に示す．

Step1：末尾文節の展開

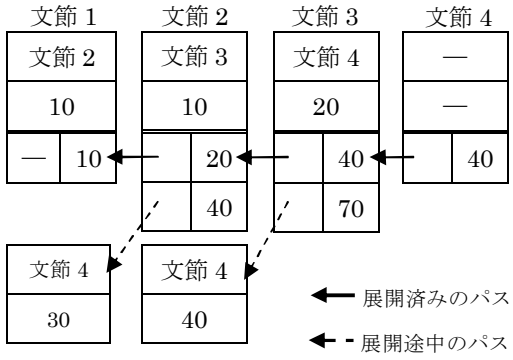
順位テーブルが空のため，初期値として末尾文節の seg 構造に対して，部分経路を逆方向に辿り，順位テーブルに格納する．

Step2：展開する解の選択

順位テーブルの先頭（コスト最小）の解候補を取り出し，Step3 に進む．

Step3：解判定

選択した解候補の係り先が先頭文節まで，決定していない場合は，Step 4 に進む．先頭文節まで決定している場合，その経路はすべての係



係り文節	4	3	2	1	合計コスト
受け候補文節 1	-	4	3	2	40
受け候補文節 2	-	4	3	4	60
受け候補文節 3	-	4	4	-	70

図 3：最小コストのパスを展開した例

り受け制約を満たしているので，これにより得られた経路を解とする．ここで，最尤探索をおこなう場合，任意の N 個の解を満たすまで，Step2 に戻る．コストの小さい順に N 個の解が得られたら解探索を終了する．

Step4：部分経路の展開

展開する任意文節の seg 構造に対して，前向きに作成された部分経路を逆向きに辿る．このときの最小コストは，式(2)により与えられる．

Step5：係り受け条件の適用

新たに展開する部分経路に対して，非交差条件，一文一格の制約などの係り受け条件を適用する．これは，新たに経路に繋がる seg 構造に対して，これまで展開している部分経路と条件判定をおこなう．非交差条件は，新しい seg 構造の係り位置と受け位置を利用し，交差する経路の有無を判定する．一文一格の制約では，受け位置の文節がすでに，同じ格を受けているかを判定する．

Step6：順位テーブルに格納

展開した経路のうち，係り受け条件を満たさなければ，順位テーブルに受け候補情報を格納しない．係り受け条件を満たす場合は，式(3)により与えられる合計コストによって，順位テーブルに昇順に格納する．

4. 評価実験

係り受け解析の評価実験には，京大コーパス[5]を用いた．本手法で形態素解析に用いたエンジンと，コーパスの文節区切りが異なる文については，人手で解答の修正をおこなった．

係り受け解析をおこなうために作成した係り受け辞書は，文節間属性情報を用いた約 200 パターンと，格フレームの約 20 万パターンで構成されている．

評価実験に用いたコーパスは，1,380 文(10,802 文節)であり，1 文平均 7.83 文節である．解析実行環境は，CPU:Pentium4 3.0GHz，OS:WindowsXP 上で稼動している．

評価は，従来の構文解析器(KNP, Cabocha)との，同文入力による解析速度の比較によりおこなった．

4.1 評価

表 1 に本手法と従来手法との解析時間を示す．表 1 より，従来手法に比べ，解析速度を約 1.4～1.8 倍程度に高めることができた．

表 1：従来手法との解析速度比較

解析手法	解析時間（1 文平均）[ms]
KNP	15.94
Cabocha	12.59
本手法	8.84

しかし，平均解析時間を上回る約 100 文(1 文平均 9.42 文節)を対象とすると，1 文平均で約 11.7[ms]という結果が得られた．これらの文は，パターン数の不足とコスト設定のため，多くの解候補が制約を満たしていなかった．つまり，上位の解候補が解と成らないため，探索する経路数が増加する．このため，解析時間が掛かったと考えられる．対策としては，係り受け関係を同定するパターン数を増加することや，コスト設定法などが考えられる．

5. おわりに

本稿では，複数の係り受け関係候補からコスト最小の解を探索する手法について述べた．コスト最小の解候補から順に，係り受け条件を判定し，N-best の解を探索する手法である．

本手法は，2 文節間にコストなどの数値的情報があれば，用いることができるため，文節間接続の統計的情報を利用して，コストを決定することを今後の課題としたい．また，係り受け条件の判定手法に関して，条件判定の時間を減らし，さらなる高速化を図りたいと考えている．

参考文献

[1] Sadao Kurohashi and Makoto Nagao：“KN Parser：Japanese Dependency/Case Structure Analyzer”，Workshop on Sharable NL Resources(1994)

[2] 工藤拓，松本祐治：“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(2002)

[3] 日本語電子化辞書「日本語語彙体系」，http://www.ntt-tec.jp/technology/C404.html

[4] 永田昌明：“前向き DP 後向き A*アルゴリズムを用いた確率的日本語形態素解析システム”，自然言語処理研究会報告(1994)

[5] 京都テキストコーパス，http://www.kc.t.u-tokyo.ac.jp/nl-resource/corpus.html