一般化LR法を用いた形態素解析と構文解析の統合

1B−6

相澤道雄 田中穂積 徳永健伸
(東京工業大学)

1 はじめに

形態素解析は通常、辞書とのマッチングをおこない単語の候補を生成し、これらの隣接する単語間の接続可能性を調べることにより、可能な候補をつくりだしている。しかし、形態的な情報だけで結果を一意に決めることは難しく、構文的な情報、意味的な情報、文脈的な情報が必要となる。これらの各レベルの処理を順番におこなう方法では、そのレベルの処理で解消できない曖昧な候補はすべて保持して、次のレベルの解析に渡さなければならない。しかし、曖昧な部分を文末まで残しておくと、候補の数は組合せ的に増えてしまう。処理を統合すると、早期に上のレベルの情報を用いることができるため、候補をしぼり込むことが可能となる。

本稿では、形態素解析と構文解析の統合について、一般化LR法の枠組を用いて実現する方法を提案する。

2 一般化LR法を用いた処理の統合

2.1 統合化への問題

本来文法の制約(構文的制約)を加える構文解析器に、さらに接続可能性の制約(形態的制約)を扱わせるために、以下の二つの問題を解決しなければならない。

一般に、構文解析で用いる品詞よりも、接続可能性の検査で用いる品詞(以下これを細品詞と呼ぶ)の方が分類が細かいので、細品詞をpreterminalにする必要がある。よって構文解析用の文法をそのまま使うことができなくなり、

(1) 細品詞と構文解析用の文法をつなげる文法(細品詞用の文法)が必要になる。

また文脈自由文法の枠組では、部分木を越えた単語間の制約を記述することができない。よって、すべての単語間の接続可能性の制約を、文脈自由文法で表現するためには、各文法記号を接続属性をもった新しい記号に換え、文法を展開することにより、すべての場合を記述しなければならない。しかし、これは文法の数が組合せ的に増え、現実的ではない(第3節参照)。

A merger for morphological and syntactic analysis using generalized LR parser.
AIZAWA Michio, TANAKA Hozumi, TOKUNAGA Takenobu
Tokyo Institute of Technology

よって

(2) 文法以外の方法で接続可能性の制約を加えなければならない

2.2 一般化LR法を用いた解決

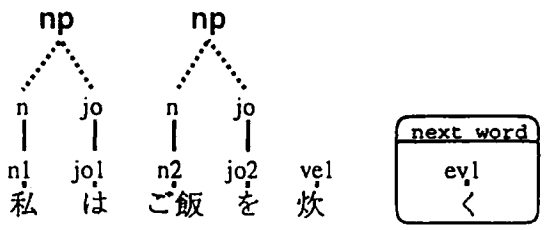

図1 解析途中の木

図1は解析途中の木で、入力文を左から右へボトムアップに解析を進め、[ve1, 炊]をシフトしたところである(図において、n1、jo1 ...が細品詞、n、joが品詞)。ここでもし、

(3) 記号ve1を含む規則がve→ve1のただ一つしかない。

ならば、次に起こる動作は規則ve→ve1の適用(レデュース動作)である。LR法では効率の良い解析をおこなうため、次の単語[ev1, く]を見、ev1がvのfollowとなっているときのみ規則を適用し、followとなっていない場合は、そこで解析が失敗する。

このとき規則を適用する条件に、

(4) ve1とev1が接続可能である。

を加えると、(3)が満たされている場合には、通常のLR解析により接続可能性の検査をおこなうことができる。つまり、次の単語と接続不可能な場合は、(4)の規則により解析が失敗する。

(1)の問題点は、(3)の条件を満たす細品詞文法を作ればよい(2.3.1参照)。(2)については、LRテーブルの作成に(4)の条件を加えることで、接続可能性の検査が可能になる(2.3.2参照)。

2.3 アルゴリズム

2.3.1 辞書・文法

細品詞用の文法は以下の細品詞規則の集合とする。

X→x (x は、品詞 X の細分類)

この結果、すべての細品詞は、ただ一つの規則に含まれる。

2.3.2 LR テーブル

まず、構文解析用の文法に細品詞用の文法を加えた文法から LR テーブルを作成する。この LR テーブルに接続可能性の検査を加えるために、以下の処理をおこなう。

- 細品詞 x を含む規則 X→x をレデュースする、テーブル中のすべてのエントリに対して、x がその先読み語と接続不可能ならば、そのエントリを除去する。

これにより (4) の規則を適用することができる。

2.3.3 基本アルゴリズム

入力を $c_1c_2 \ldots c_N$ とする (c_iは文字)。c_iと c_{i+1}の間の位置番号を i と決める。初期状態は、位置 0 に状態 0 のノードをひとつ置く。i=0 から N まで、以下の処理をおこなう。i=N でアクセプト動作が起これば解析に成功する。

1. 辞書を引き、c_{i+1}から始まる単語をすべて求め、これを先読み語とする。
2. 位置 i にある全てのノードに対し、それぞれの先読み語に対する解析をおこなう。シフトがおこったスタックは、トップの位置を i+l とする (l は単語の長さ)。全てのスタックのシフト動作が終れば、文字位置 i での処理は終る。

2.3.4 マージのタイミング

一般化 LR 法では、シフトの前にマージをおこなう。しかし本手法では、すべてのスタックのシフト動作が終了してからマージをおこなう必要がある。その理由を簡単に述べる。接続属性の情報を持っている文法記号は細品詞だけであり、細品詞がスタックトップにあるのは、シフト動作の直後である。その他のタイミングでは、スタックトップが接続属性の情報を持っていない。スタックトップに接続属性の情報がない時に、マージをおこなうと、接続可能性の制約をうまくかけることができなくなる。よって、マージはシフト動作の後におこなわなければならない。

3 テーブルの大きさについての考察

文法 1 は規則数 48 の日本語の文法である。本手法を用いた場合のテーブルの大きさを次にしめす。

	規則	細品詞	非終端	状態	エントリ
文法 1	48	-	15	54	1166
本手法	485	437	45	491	28354

次に文法を展開する方法と比較する。文法 2 と文法 3 は、文法 1 の一部を用いている。

verb → ve, ev.

文法 2

	規則	細品詞	非終端	状態	エントリ
文法 2	1	-	1	3	4
展開	288	175	175	329	1236
本手法	176	175	3	178	944

pred → verb, jds.
verb → ve,ev.
jds → jd.
jds → jd, jds.

文法 3

	規則	細品詞	非終端	状態	エントリ
文法 3	4	-	3	7	13
展開	3815	225	580	3932	13946
本手法	229	225	6	232	2253

文法 2 と文法 3 の結果からわかるように、展開する方法はすぐに状態数が爆発してしまい、現実的ではない。本手法では状態数は、細品詞を加えない文法に比べて、細品詞規則の数だけしか増えないため、テーブルのサイズを抑えることができる。

4 おわりに

一般化 LR 法を用いた、形態素解析と構文解析の統合方法を提案した。

今後は、意味的な情報などを採り入れるための枠組を考える予定である。

謝辞

研究を進めるにあたり EDR の日本語単語辞書評価版を使わせて頂きました。また、接続テーブルを下さった富士通の内田裕士さんに感謝いたします。

参考文献

[1] 日本電子化辞書研究所:TR-018 日本語単語辞書, 日本電子化辞書研究所.

[2] 沼崎浩明:並列計算モデルを用いた自然言語処理の高速化に関する研究, 東京工業大学博士論文,(1992).

[3] Tomita,M.: An Efficient Augmented Contex Free Parsing Algorithm., Computational Linguistics,Vol 13,Number1-2,pp314-46(1987).