構文解析情報を利用した英語品詞列選定

2R－7

吉村 裕美子
(株)東芝　研究・開発センター

1　はじめに

構文解析精度を向上するためには、全体の構文解析率を上げながらそのうちの正しい品詞列に対する成功の割合を下げないことが必要である。それには、より多様な表現を構文解析できる能力の向上と、正しい品詞列を導ける能力の向上の2つの取り組み([1],[2]等)がある。本稿では、前者の能力は固定した上で構文解析知識にかなう文に対していかに正しい品詞列を効率よく導くことができるかという英語品詞列選定の制御について述べる。

2　品詞列選定処理の概要

本稿で取り上げる品詞列選定処理は、以下のような処理手順からなっている。

(1) **優先・非優先条件判定** 優先・非優先規則により、ある部分単語列に対する品詞列の優先度を高める、あるいは低める処理をする。優先度は、基本的に語と語の隣接に対して付与し、一文全体の系列に対する優先度は設定していない。

(2) **品詞列展開** 優先・非優先条件判定結果を元に、最も優先度の高い系列の選択を行う。

(3) **禁止条件判定** 選択された一系列に対して、禁止条件判定規則により、構文解析にかける前に明らかに誤った品詞列を含まないかどうかを判定し、誤品詞列は棄却する。棄却後は(2)に戻り、次に優先度の高い系列の選択処理に入る。

(1)、(2)、(3)の処理を経て、ある一つの品詞列が出力される。これが続く構文解析過程の入力となる。構文解析において、系列が構文解析知識では受理できず失敗となった場合は、(2)へ戻り、次に優先度の高い品詞列候補を選択する。これを、構文解析が成功するまで、あるいは品詞列候補が尽きるまで行う。

3　評価基盤

正しい品詞列を導ける選定精度を判定するために、13,543文に対して正しい品詞情報をタグづけした正解データを作成した。内訳は次のとおりである。

(a) UNIXオンラインマニュアル	5,525文
(b) ネットワーク技術文書	5,159文
(c) 雑誌記事	2,859文

上記データに相当する文書の70%の文に対して、実験用の構文解析規則を使い、最初に構文解析部に送られてくる品詞列と、構文解析を成功する品詞列の正解率を、単語単位と文単位とで算出した。構文解析に成功するまで、最高15種類の異なる品詞系列を展開するものとした。表1にその結果を示す。

文書	第1品詞列 正解数[A]/総数[B]	構文解析成功品詞列 正解数[C]/成功総数[D]
(a)	49,296/50,538(97.54%)	31,417/31,974(98.26%)
	3,062/3,946(77.60%)	2,546/2,942(86.54%)
(b)	62,406/63,114(98.88%)	39,272/39,677(98.98%)
	2,792/3,324(84.00%)	2,124/2,411(88.10%)
(c)	41,629/42,365(98.26%)	20,020/20,296(98.64%)
	1,822/2,420(75.29%)	1,118/1,344(83.18%)

表1: 基盤となる精度
(上：単語単位、下：文単位)

構文解析規則は万能ではないため、規則として記述されている知識では受理できない文が存在する。さらに、解析成功品詞列の精度が100%でないことから、知識の不足のために誤った品詞列をも棄却しきれずに11-14%程度受理していることがわかる。一方、構文解析知識は備えているが、正しい品詞列が展開されるまでに、制限値である15種類に至ってしまい、結果として構文解析を成功しない文も存在することが予想される。構文解析規則の知識が固定であり、規則にかなうような解析結果が得られる割合がほぼ一定と仮定すると、構文解析を成功する文の割合を向上させることが、全体として正しい解析を得られる割合の増加（＝[C]の数の増加）につながると言える。そこで、特定の品詞列に対して構文解析の失敗が明らかになった時点で次の品詞列候補を選択する際に、構文解析規則の知識にかなう品詞列をいかに効率良く導くことができるかに焦点を当てることとした。

Selection of English part-of-speech strings using syntactic analysis information
Yumiko YOSHIMURA
Toshiba Corporation

4 解析不成功情報の利用

禁止条件判定および構文解析処理において特定の品詞列が棄却された際には、禁止条件判定規則が棄却を判定した語、および構文解析不成功が明らかになった時点までに解析処理が読み進めていた語より後ろ方向（文頭方向）の語の中に、誤った品詞が割り振られた語が存在する。表1に示した評価は、その範囲の中に最低1つの変更されるべき誤り品詞があるものとして、次候補品詞列を選定するように設定していた。（変更した語に連動してその範囲外の語の品詞も変更されることは当然ある。）そこで、試みとしてその範囲を種々に変更して解析成功する文数がどのように変化するかをみることにした。具体的には、棄却時に処理を進めていた先端語から4語～8語を範囲としてそれぞれ評価を行った。結果を表2に示す。

範囲	文書 (a)	文書 (b)	文書 (c)
4語	36,613/37,251	44,349/44,777	21,303/21,592
	2,767/3,202	2,282/2,589	1,170/1,406
5語	36,568/37,209	44,255/44,685	21,222/21,506
	2,767/3,202	2,280/2,589	1,170/1,405
6語	36,662/37,306	44,426/44,856	21,322/21,610
	2,768/3,205	2,282/2,591	1,170/1,406
7語	36,649/37,293	44,253/44,684	21,239/21,524
	2,767/3,204	2,276/2,586	1,169/1,405
8語	36,629/37,271	44,284/44,716	21,251/21,540
	2,767/3,203	2,276/2,587	1,169/1,406
制限	31,417/31,974	39,272/39,677	20,020/20,296
無し	2,546/2,942	2,124/2,411	1,118/1,344

表 2: 範囲語数別の解析成功品詞列精度の変化

（上：単語単位、下：文単位）

表2から明らかなように「範囲6単語」で正しい品詞列に対して構文解析を成功する量がピークとなっている。この実験に際して、指定範囲の中に品詞を変更できる別候補が存在しないとき、あるいは別候補に変更した品詞列がすべて以降の処理で棄却された場合には、展開する品詞列数の制限値15に至らなくても、その文に対する品詞列選定・構文解析処理は失敗とするという制約を付与した。この制約のために正しい品詞列が展開しえなかった文が存在したはずだが、結果としては制約のない表1の結果より精度が高くなっている。表1の場合は、制限値に至るまえに正しい品詞列を展開できない文が存在したことがわかる。つまり、優先・非優先条件判定結果より制約の付与のほうが確からしい品詞列を展開する上では効果があった。また、禁止条件判定規則は隣接3語以内に関するものがほとんどであることから、この規則による棄却情報が主に生かされるのなら「範囲4単語」の精度が高いはずである。実際「範囲5単語」より精度は高い。しかしピークにはなっていないことから、構文解析規則による棄却情報がもっと広い範囲で効果を見せていると推測される。

表2の結果から、正しい品詞列が構文解析を成功する量をより上げるには、「範囲6単語」の中で誤り品詞を選定するのを優先して品詞列展開を行い、制約内では展開すべき品詞列がなくなったら、範囲を削除して展開品詞列数の制限値まで次候補品詞列の展開を行うようにするのが良いと判断できる。そこで、簡単な検証のために、通常の制約内の品詞列展開で構文解析を成功しない場合は、制約をなくして再度品詞列展開・構文解析をしなおすという仕様で実験を行い、精度の変化を出力してみた。表3にその結果を示す。

単位	文書 (a)	文書 (b)	文書 (c)
語	36,820/37,468	44,777/45,217	21,488/21,791
文	2,773/3,213	2,294/2,610	1,173/1,418

表 3: 改良仕様での精度

5 おわりに

構文解析の失敗が明らかになった時点での処理状況をもとに、次の品詞列の候補を選定する際に、品詞の変更を行う語の範囲を適度に制限することで、構文解析を成功する正しい品詞列をより早く導けることを示した。

英語の品詞タグ付けの精度は単語単位では97-98%代と高い数値となるが、文単位に換算すると70-80%代となる。正しい構文解析結果を得るにはもっと精度を追及する必要があることがわかる。誤り品詞列を認識する上では、個々の品詞列に対する構文解析（部分）木の妥当性を評価することも有効である。今後は、このような品詞列の不確からしさを認識する能力についても検討していきたい。

参考文献

[1] Pasi Tapanainen and Atro Voutilainen, "Tagging accurately - Don't guess if you know", Proceedings of Fourth Conference on Applied Natural Language Processing, 1994

[2] Eric Brill, "A Simple Rule-Based Part of Speech Tagger", Proceedings of Third Conference on Applied Natural Language Processing, 1992