自然言語処理　59－1
（1987. 1. 23）

効率的な英文解析のための多品詞解消方式

小原　和博

ＮＴＴ情報通信処理研究所

本論文では、効率的な英文解析を行なうために有効な多品詞解消方式を新たに提案するとともに、科学技術英文を対象とした多品詞解消実験による評価結果ならびに英文解析時間の短縮効果について述べる。　主な結論は以下のとうりである。
（１）多品詞解消能力：　１００文を対象とした多品詞解消実験を行なった結果、本論文で提案する多品詞解消方式により、①１単語当り１．２７の多品詞性が、出力時には１．１０の多品詞性に解消できることがわかった。②そのときの正解率は９８．６％であった。
（２）英文解析時間の短縮効果：　入力英文の品詞の組合せごとに句構造ルールとの照合を図る構文解析手法を前提とした場合、①多品詞解消モジュールを利用したときの英文解析時間の短縮効果は単語数に対してほぼ指数関数的に増大する可能性がある。②２３単語(試験文の平均単語数)の英文を解析する場合には、約２０倍速く英文解析できる可能性がある。

Resolving Categorial Ambiguities for Efficient English Sentence Parsing

Kazuhiro Kohara

NTT Communications and Information Processing Laboratories
1-2356 Take Yokosuka-Shi Kanagawa 238-03 Japan

This paper presents not only a new algorithm for resolving categorial ambiguities(multiple parts of speech) but also its capability given by an evaluation experiment on 100 sentences. Its qualitative effects on parsing time are also discussed, assuming a parsing method pursuing all categorial ly possible parse paths in breadth-first fashion. Main results are as follows.
(1)According to the categorial disambiguation experiment, the number of POS per word can be reduced from 1.27 to 1.10. The correctness of the disambiguation is 98.6%.
(2)According to the estimation of shotening effects, the processing speed by the parser with the categorial disambiguation module can be about 20 times faster than that without the disambiguation module, in the case of parsing a 23 word sentence.

1　はじめに

自然言語理解や機械翻訳などの英文解析では、品詞のあいまい性、句や節の係り受けのあいまい性など様々なあいまい性の解消が重要な問題となる。このうち本論文では多品詞性の解消の問題について議論する。

英単語のもつ品詞のあいまい性のために、英文解析では品詞の組合せ数の爆発的増加に直面する。　そこでこの問題の解決策として、構文解析モジュールと独立した多品詞解消モジュールを用意しておき、これにより予め品詞の数を効率的に減少する方法が考えられる。

これまで自然言語の構文解析に関して、アルゴリズム、言語、文法といった様々な側面から多くの研究がなされてきた[3]～[10]。　しかし多品詞解消の研究は、英文解析時間の短縮に極めて有効であるにもかかわらず、これまで余り研究されていない[1]。　数少ない研究報告[1][2]はあるものの、多品詞解消能力の十分な評価や、英文解析時間の短縮効果への定量的評価がなされていない。

そこで本論文では、効率的な英文解析を行なうために有効な多品詞解消方式を新たに提案するとともに、科学技術英文を対象とした多品詞解消実験による評価結果ならびに英文解析時間の短縮効果について述べる。

2　多品詞解消方式

本論文で扱う品詞の分解能は以下に示す１３品詞である。

(1)名詞、(2)代名詞、(3)関係代名詞、(4)冠詞、(5)形容詞、(6)副詞、(7)関係副詞、(8)前置詞、(9)接続詞、(10)be動詞、(11)助動詞、(12)自動詞、(13)他動詞

各品詞には必要に応じて適当なプロパティを用意する。　例えば、形容詞には限定用法だけに用いられる形容詞か否か、他動詞には「ｅｄ形」(例：resolved)か「ｉｎｇ形」(例：resolving)か否か(例：resolve)というプロパティを付与する。

２．１　多品詞解消ルール

多品詞解消ルールの例を図１に示す。実際のルールは２３×２３（動詞にはｅｄ形、ｉｎｇ形の欄を追加し、品詞の他にカンマ、ピリオド、文頭の欄を追加）のマトリックスから成る。

多品詞解消ルールには、いかなる場合にも該当する品詞の並びを許さない無条件ルール（図１で×で示したルール）と、該当する品詞の並びに対して特別の解消条件を与える条件ルール（図１で数字で示したルール）がある。

A＼B		名詞	冠詞	自動詞			他動詞			ピリオド
					ing	ed		ing	ed	
冠詞			×	×		×	×			×
形容詞			22	×		×	×			9
自動詞		14	14	×		×	×			
	ing									
	ed	14	14	×		×	×			
他動詞				×		×	×			×
	ing									
	ed									

図１　多品詞解消ルールの例

無条件ルールは全部で１１１箇所にあり、条件ルールは全部で６２箇所にある。

ｅｄ形でもｉｎｇ形でもない動詞に関する無条件ルールの例を以下に示す。

①自動詞あるいは他動詞は冠詞の直後に来てはいけない。
②自動詞あるいは他動詞は、自動詞あるいは他動詞の直前に来てはいけない。
③他動詞はピリオドの直前に来てはいけない。

単語Ｂが単語Ａの直後にある場合の、条件ルールの例を以下に示す。

［条件９］単語Ａが限定用法だけに用いられる形容詞を含む多品詞語であり、しかもそれがピリオドの直前にある場合には、単語Ａから形容詞を削除する。
［条件14］単語Ａが多品詞性のない完全自動詞であり、単語Ｂが名詞を含む多品詞語の場合には、単語Ｂから名詞を削除する。　単語Ｂに多品詞性がなく、しかも単語Ａが完全自動詞を含む多品詞語の場合には、単語Ａから自動詞を削除する。
［条件22］単語Ａが形容詞を含む多品詞語（allとsuch以外）であり、単語Ｂが冠詞の場合には、単語Ａから形容詞を削除する。　単語Ａがallかsuchであり、単語Ｂが冠詞の場合には単語Ｂの品詞を形容詞に決定する。

解消ルールは直後の多品詞性を解消するための前方ルールと、直前の多品詞性を解消するための後方ルール、両方向の多品詞性を解消するための両方向ルールに分れる。　また解消ルールは一定の品詞を削除するための禁止ルールと、一定の品詞を選択するための選択ルールに分れる。　無条件ルールは全て禁止ルールである。

２．２　多品詞解消アルゴリズム

多品詞解消アルゴリズムの主要部を以下に示す。

［ＳＴＥＰ１］ｉ番めの入力単語の品詞の数Ａin(ｉ)を求める。
［ＳＴＥＰ２］Ａin(ｉ)の積Ｓを求める。
［ＳＴＥＰ３］入力順にＡin(ｉ)＞１とＡin(ｉ＋１)＝１の２単語列を検出して、その単語列に該当する後方ルールと両方向ルールを適用する。
［ＳＴＥＰ４］入力順にＡin(ｉ)＝１とＡin(ｉ＋１)＞１の２単語列を検出して、その単語列に該当する前方ルールと両方向ルールを適用する。
［ＳＴＥＰ５］Ｓを更新する。
［ＳＴＥＰ６］Ｓが減少し、かつ、全てのＡin(ｉ)がまだ１でない場合には[STEP3]からの処理を繰り返す。　その他の場合には多品詞解消処理を終了する。

本アルゴリズムの特徴は次のとうりである。

（１）多品詞性のない単語と多品詞性のある単語が並んでいる単語列を検出して、その部分に多品詞解消ルールを適用する。　これにより制御が簡単で効率の良い多品詞解消処理を実現できる。
（２）必ずしも品詞を１つに決定することはせず、極めて正解率の高い解消ルールにより、品詞の数を無理なく減少させる。　これにより修復のためのメカニズムが不要となりさらに制御が簡単になる。

3　多品詞解消実験

3．1　実験方法

科学技術文献から抽出した１００文を対象に、多品詞解消アルゴリズムを机上でトレースすることにより多品詞解消実験を行なった。

図２に多品詞解消の実験例を示す。この場合、thenの形容詞と、followsの他動詞が後方ルールにより削除され、chipの他動詞が前方ルールにより削除され、chipの自動詞、designの自動詞と他動詞が両方向ルールにより削除される。　品詞数の積は３６通りから１通りに減少し、また、１単語当りの品詞数の相乗平均は１．８２個から１個に減少する。

入力英文	入力の品詞(Ain)	出力の品詞(Aout)
Then	adv,adj(2)	adv (1)
the	article(1)	article(1)
chip	n,vt,vi(3)	n (1)
architecture	n (1)	n (1)
design	n,vt,vi(3)	n (1)
follows	vt,vi (2)	vi (1)
.	period (1)	period (1)
品詞数の積	(36)	(1)
品詞数の相乗平均	(1.82)	(1.00)

図２　多品詞解消実験例

3．2　実験結果

表１に実験データを示す。　入力文の単語数をｎ、入力時の品詞数の相乗平均をＡin(av)、出力時の品詞数の相乗平均をＡout(av)とすると、試験文１００文での相加平均は以下のとうりである。

ｎ＝２２．９単語
Ａin(av) ＝１．２７品詞
Ａout(av)＝１．１０品詞

これらの値は文献１と文献２とで余り差がない。

次に、多品詞解消アルゴリズムにより品詞を削除した件数と、正解品詞を削除してしまった件数とからアルゴリズムの正解率を求めると、全体で９８．６％となる。　この値も文献１と文献２とで余り差がない。

以上から平均的にみると、２２．９単語の英文を多品詞解消モジュールに入力した場合、１単語当り１．２７の多品詞性が、出力時には１．１０に多品詞解消されることがわかる。　また、そのときの正解率は９８．６％である。

多品詞性の残りやすいポイントは以下のとうりである。

①カンマの直前の多品詞語
　例：・・・ design, ・・・
②(前置詞と接続詞)の多品詞語
　例： as after
③(形容詞と副詞)の多品詞語
　例： more high even
④thatと、動詞を含む多品詞語が並んでいる単語列
　例：・・・a prompt that matches・・・
⑤ｅｄ形／ｉｎｇ形の動詞
　例：・・・algorithms tailored to・・・

表１　多品詞解消実験データ

	文献１(50文)	文献２(50文)	全体(100文)
平均単語数(単語)	２３．２	２２．７	２２．９
入力時の品詞数の相乗平均(品詞)	１．３１	１．２４	１．２７
出力時の品詞数の相乗平均(品詞)	１．１１	１．１０	１．１０
品詞を削除した件数ｘ(件)	２９６	２６０	５５６
正解品詞を削除した件数ｙ(件)	３	５	８
正解率（ｘ－ｙ）／ｘ（％）	９９．０	９８．１	９８．６

[文献１] Muroga,S.：VLSI System Design,John Wiley & Sons(1982)
[文献２] Abraham,E et al.：The Optical Computer,Scientific American(Feb.1983)

これらの多品詞性は、２単語間の接続関係を中心とした解消ルールでは解消が困難なものばかりである。　中には３単語以上のスコープを見る解消ルールにより多品詞解消できる場合もあるが、ほとんどは本格的な構文解析により解消されるものである。

副作用の生じた主なポイントは以下のとうりである。

①接続詞と接続詞の単語列
　例：But if we want to･･･
　（Butの接続詞を削除）
②前置詞と前置詞の単語列
　例：･･･are referred to as the･･･
　（asの前置詞を削除）
③現在形他動詞とｔｏ以外の前置詞の単語列
　例：･･･require from 1 to 10･･･
　（requireの他動詞を削除）
④文末にある原形他動詞を削除
　例：･･･is difficult to change.
　（changeの他動詞を削除）

これらの副作用を除く方法として、解消ルールの制限条件を強化する案と、解消ルールを削減する案が考えられる。しかし、前者の案では解消時間が増大するという問題が生じ、後者の案では多品詞解消能力が低下するという問題が生じる。　これらの問題はトレードオフの関係にある。

４　英文解析時間の短縮効果

実験結果に基づき、多品詞解消モジュールを利用した場合の英文解析時間の短縮効果について試算してみる。但し、構文解析の手法として、入力英文の品詞の全ての組合せに対して句構造ルールとの照合を図る方法(Breadth-First Parsing[10])を想定する。

４．１　短縮効果の理論式

多品詞解消モジュールへの入力時の品詞の組合せ数（各単語の品詞数の積）をｊ、出力時の品詞の組合せ数をｋと

し、j通りの構文的あいまい性をもつ英文の構文解析時間をf(j)、多品詞解消時間をg(j)とする。　このとき、解消モジュールのない場合の英文解析時間T0と、ある場合の英文解析時間T1は以下のようになる。

$$T0 = f(j) \quad (1)$$

$$T1 = g(j) + f(k) \quad (2)$$

構文的に正しい品詞をもつ英文の構文解析時間をf0とする。　その他の品詞の組合せに対する構文解析時間を(1/m)f0とすると（一般的に言って、m>1）、f(j)、f(k)は以下のようになる。

$$f(j) = (j-1)(1/m)f0 + f0 \quad (3)$$

$$f(k) = (k-1)(1/m)f0 + f0 \quad (4)$$

一般的に言って、g(j)はf(j)よりも小さい。

$$g(j) = a \cdot f(j) \quad (5)$$

$$0 < a < 1 \quad (6)$$

但し、aは定数とは限らない。

多品詞解消モジュールによる英文解析時間の短縮効果Eを以下のように定義する。

$$E = T0 / T1 \quad (7)$$

式(1)～(5)からEは次のようになる。

$$E=(j+m-1)/[a(j+m-1)+(k+m-1)] \quad (8)$$

4．2　短縮効果の推定値

入力英文の単語数をnとしたとき、表1に示した実験結果からjとkの値を以下のように考える。

$$j = 1.27^{n} \quad (9)$$

$$k = 1.10^{n} \quad (10)$$

式(9)(10)を式(8)に代入すると

$$E=(1.27^{n}+m-1)/[a(1.27^{n}+m-1)+(1.10^{n}+m-1)] \quad (11)$$

ある正しい品詞をもつ英文の構文解析時間をf0としたとき、他の品詞の組合せに対する解析時間を、途中で失敗しようが別の解析木を作ろうが平均的に(1/2)f0と仮定する（m＝2）。n=22.9（平均単語数）とm=2を式(11)に代入して求めたEとaの関係を図3に示す。

実際に設計中の多品詞解消プログラムと構文解析プログラムを見て、f0とgの動的ルール数を机上で見積り、動的ルール数が解析時間に比例すると仮定して、以下の式によりaを求めた。

$$a = g/[(j-1)(1/2)f0 + f0] \quad (12)$$

その結果、aは高々0.008であることがわかった。　a=0.008とm=2を式(11)に代入して、Eとnの関係をプロットしたものを図4に示す。　図4から次のことが言える。

①英文解析時間の短縮効果は単語数に対してほぼ指数関数的に増大する可能性がある。

②23単語（試験英文での平均的な単語数）の英文を解析するとき、多品詞解消モジュールを利用することにより約20倍速く英文解析できる可能性がある。

5　おわりに

効率的な英文解析のための多品詞解消方式について述べた。　主な結論は以下のとうりである。

（1）多品詞解消能力

100文を対象とした多品詞解消実験を行なった結果、本論文で提案した多品詞解消方式により、

①1単語当り1．27の多品詞性が、出力時には1．10の多品詞性に解消できることがわかった。

②そのときの正解率は98．6％であった。

（2）英文解析時間の短縮効果

入力英文の品詞の組合せごとに句構

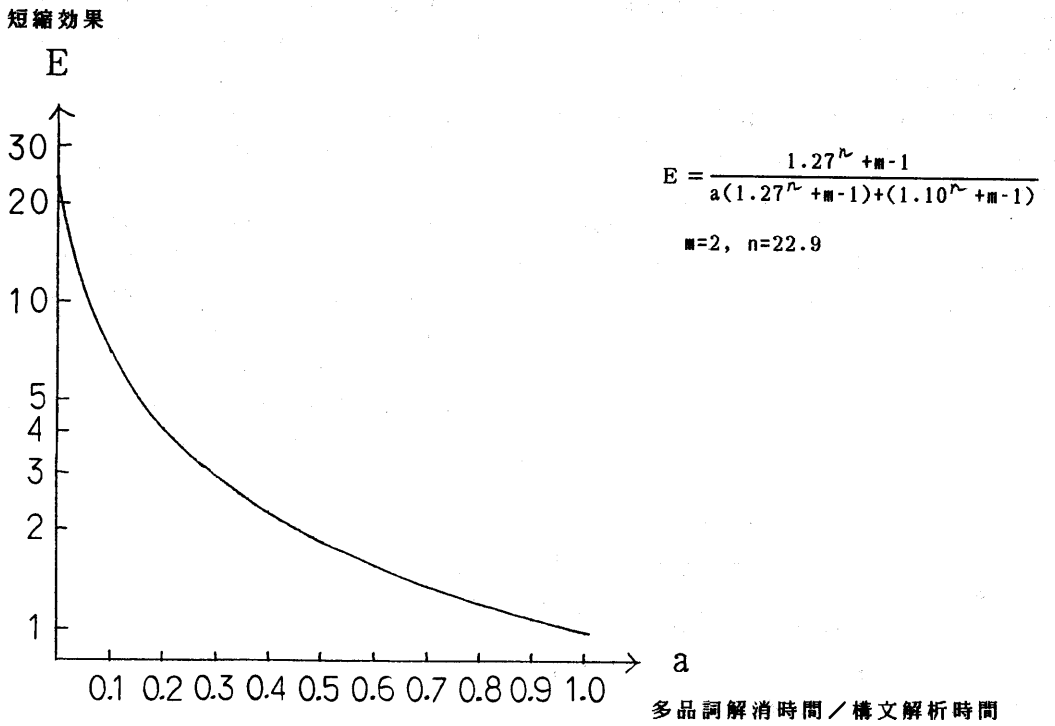

図３　英文解析時間の短縮効果１

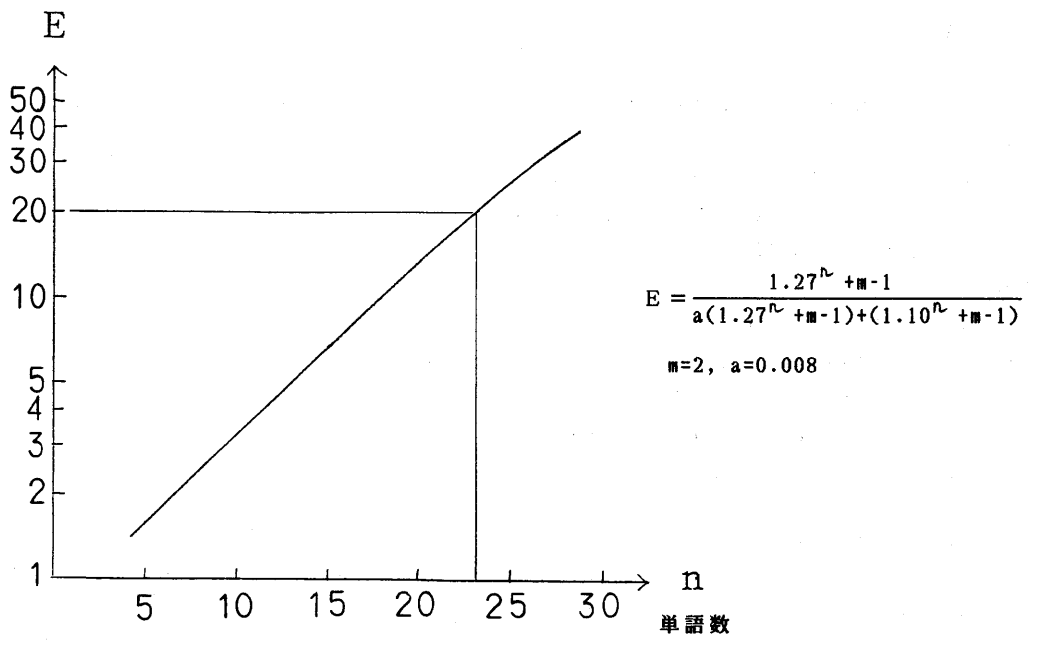

図４　英文解析時間の短縮効果２

造ルールとの照合を図る構文解析手法を前提とした場合、
①多品詞解消モジュールを利用したときの英文解析時間の短縮効果は単語数に対してほぼ指数関数的に増大する可能性がある。
②２３単語(試験文の平均単語数)の英文を解析する場合には、約２０倍速く英文解析できる可能性がある。

謝辞

日頃御指導いただく阿部豊彦研究室長に感謝します。　本論文に関して貴重な御討論をいただいた下村二三男主幹研究員、片桐恭弘主任研究員はじめ関係各位に感謝します。　多品詞解消実験に関して多大なる御協力をいただいた菊井玄一郎君、三浦裕壮君、水嶋いづみ嬢に感謝します。

参考文献

[1]Duffy,G.:Categorial Disambiguation, Proc. AAAI'86(1986)
[2]山本利文、辻井潤一、長尾真:Muプロジェクトの英日機械翻訳システムにおける多品詞語の解消, NL研53-7(1986.1)
[3]Aho and Ullman:The Theory of Parsing, Translation and Compiling, Vol.1, Prentice Hall(1975)
[4]Earley,J.:An Efficient Context-free Parsing Algorithm, CACM 6(1970)
[5]Tomita,M.:An Efficient Context-free Parsing Algorithm for Natural Languages, Proc. IJCAI'85(1985)
[6]Pratt,V.R.:A Linguistic Oriented Programming Language, Proc. IJCAI'73(1973)
[7]Tanaka H. et al.:Predictive Control Parser:Extended LINGOL, Proc. IJCAI'79(1979)
[8]Woods,W.A.:Transition Network Grammars for Natural Language Analysis, CACM 13(1970)
[9]Tennant,H.:Natural Language Processing, Petrocelli Books(1981)
[10]Martin,W.A. et al.:Preliminary Analysis of a Breadth-First Parsing Algorithm: Theoretical and Experimental Results, TR261, MIT Laboratory for Computer Science(1981)