自然言語処理　60－1
（1987. 3. 27）

コスト最小法を用いた日本語文の形態素解析

吉村　賢治　　武内　美津乃　　津田　健蔵　　首藤　公昭
福岡大学　　工学部

本稿では，表の広がりを多段階に制御する表方式の形態素解析アルゴリズムとその実験について報告する．このアルゴリズムでは，正しい解析結果を早く出力するためのヒューリスティックとしてコスト最小法を用いており，表の展開を制御して処理の能率を上げるために文節末の可能性の強さに関するヒューリスティックを用いている．解析実験は，コスト最小法だけを用いた表方式のアルゴリズムと，それに表の広がりを制御する機構を取り入れたアルゴリズムについて行い，解析の精度と能率を比較した．その結果，表の広がりを制御するアルゴリズムは制御しないものに比べて解析の精度は若干悪くなるが，解析に要する時間は１／３程度になることが確認できた．

Morphological Analysis of Japanese Sentences using The Least Cost Method

Kenji YOSHIMURA, Mitsuno TAKEUCHI, Kenzo TSUDA, Kosho SHUDO

Department of Electronic Engineering, Fukuoka University,
8-19-1 Nanakuma Jonan-ku, Fukuoka,814-01, Japan

An algorithm for morphological analysis of Japanese sentences is described. The algorithm is basically a tabular method, but has a mechanism to control the growth of the parse table. We employ two kinds of heuristic in this algorithm. One is a heuristic to select the correct structures named The Least Cost Method, and the other is a heuristic on a possibility of the ending of BUNSETSU(a sentence unit of Japanese) to control the growth of the parse table. By the experiment, we compared the accuracy and the efficiency of the algorithm that uses only The Least Cost Method with those of the algorithm that uses both heuristics. The accuracy of the latter is a little worse than the former's, but the latter seems to require only one third processing time of the former to analyze sentences.

１．はじめに

実用的な日本語文解析システムにおいて，入力文中に存在する未登録語の位置や品詞等の推定は不可欠な処理である．筆者等は，参考文献(1)で入力文中の未登録語に対処できる表方式の形態素解析アルゴリズムについて報告した．このアルゴリズムでは，文節数最小法[2]の考え方を一般化したコスト最小法の枠組みで，システムの単語辞書に登録されていない片仮名・アルファベット列や１文字の漢字，平仮名を単語と同等に扱うことで未登録語に対処しており，約95%の精度で未登録語の品詞を推定することができる[3]．しかし，１文字の漢字や平仮名を単語と同等に扱っているために入力文中の１文字毎に自立語辞書の検索を行うという欠点を持っている．この欠点を補うために，筆者等は文節末の可能性の強さに関するヒューリスティックを用いて解析表の展開を多段階に制御するアルゴリズムを提案した[4]．本稿では，コスト最小法における各コスト値の設定と表の展開を制御するアルゴリズムを示し，表の展開を制御するアルゴリズムと制御しないアルゴリズムについて行った解析の精度と能率に関する比較実験の結果について報告する．

２．文法モデル

２．１　日本語文の構造

アルゴリズムの記述を簡潔に行うために，ここでは次のように簡略化した日本語文の構造に基づいて議論を行う．

[1] [文]::=[文節]|[文]･[文節]
[2] [文節]::=[自立語]|[文節]･[付属語]

本稿で述べるアルゴリズムでは，[2]で示される文節の構造を規定する規則のみを用いて，[1]に示されている文の構造を規定する規則は用いていない．なお，筆者等の形態素解析システムでは[付属語]として，一般の付属語の概念を拡張した関係表現と助述表現と呼ぶ付属語的表現を用いているが，アルゴリズムに関する議論においてこれらの区別は重要ではないため，以下では単に付属語と呼ぶ．

２．２　文節の文法モデル

本節では２．１の[2]で示した文節の構造を形式的に定義する．以下の議論では長さnの文字列をsで表し，先頭からi番目の文字をs(i)で表す．つまり，s=s(1)s(2)…s(n)とする．

［定義１］　諸述語の定義
(1) 入力文字列の部分列s(i+1)s(i+2)…s(j)が文法情報αの単語であることをword(i,j,α)で表す．
(2) αが自立語の文法情報であることをcword(α)で表す．
(3) αが付属語の文法情報であることをfword(α)で表す．
(4) 文法情報αの単語W_1と文法情報βの単語W_2の接続W_1W_2が文節内で可能であることをconnect(α,β)で表す．
(5) 文法情報αをもつ単語で文節を終われることをend(α)で表す．　■

ここで，文法情報とは品詞，活用型，活用形などの情報である．

これらの述語を用いて文節の構造を規定する規則は次のように表現することができる．

［定義２］　文節の文法モデル
文字列s=s(1)s(2)…s(n)に対して，整数$i_0,i_1,\cdots,i_{m+1}$($i_0=0,i_{m+1}=n,m\geqq 0$)が存在し，
(1)word(i_j,i_{j+1},α_j)　　(j=0,1,…,m)
(2)cword(α_0)
(3)connect(α_j,α_{j+1})　　(j=0,1,…,m-1)
(4)end(α_m)
を満たすとき文字列sは文節であると定義する．ただし，m=0のとき(3)は除外する．　■

この文節の文法モデルは，文節を構成する単語の並びを隣接する２単語間の接続ルールだけで規定しているため，文節であるための必要条件にしかすぎないが，文法的に正しい日本語文を解析する場合にはこのモデルで十分である．

３．形態素解析アルゴリズム

３．１　アルゴリズム記述のための準備

本章では，２章で定義した文節の文法モデルに基づいて日本語文の形態素解析を行うアルゴリズムについて述べる．このアルゴリズムは基本的には表方式のアルゴリズムであり，入力文を文頭から文末に向かって走査して解析表を作成するアルゴリズムと，この解析表を文末側から文頭側に走査して解析結果を抽出するアルゴリズムとで構成される．

以下，アルゴリズムを簡潔に記述するために幾つかの定義を行う．

［定義３］　集合 Cset,Fset

集合Cset(i)とFset(i)を次のように定義する．

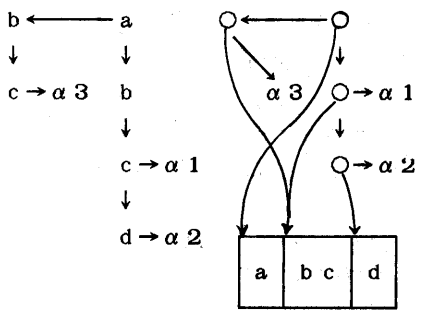
Cset(i)={(i,j,α)| word(i,j,α) & cword(α)}

Fset(i)={(i,j,α)| word(i,j,α) & fword(α)}

■

入力文字列sを与えてCset(i)またはFset(i)を求めることは，それぞれ入力文字列sの部分列s(i+1)s(i+2)…s(n)の先頭から始まる自立語または付属語をすべて求めることに対応する．ここでは単語辞書の詳細なデータ構造については触れないが，与えられた文字列の先頭から始まるすべての単語を一定の時間以内で検索できる，単語辞書に適したデータ構造として，単語を構成しているそれぞれの文字をノードとした木構造であるTRIE構造[5]がある．しかし，TRIE構造の一つのノードは，文字を格納する場所と他のノードを指す二つのポインタと辞書内容を指すポインタのための場所を必要とするため，辞書の容量が大きくなるという欠点をもつ．筆者等のシステムでは，TRIE構造を改良して単語辞書の容量を小さくする工夫をしている（図１）．

登録されている単語	辞書内容
a b c	α１
a b c d	α２
b c	α３



TRIE構造　　改良したTRIE構造

図１　単語辞書のデータ構造

３．２　ヒューリスティック

本稿で述べるアルゴリズムでは二つのヒューリスティックを利用している．

形態素解析において，複合語の分割や未登録語の処理を行うためには２章で述べた文節の文法モデルだけでは能力が弱く，この文法モデルを満足する解析結果の中には誤った解析も多数含まれている．これらの中から正しい解析を効率良く見つけるために，本アルゴリズムでは文節数最小法の考え方を一般化したコスト最小法を用いている．文節数最小法は自立語のコストを１，付属語のコストを０とした場合のコスト最小法と考えることができる．文法情報αに対して，そのコストを与える関数をcost(α)とする．ここで関数costの値は非負整数とする．どのような分類に対して，いくらの値を与えるかなどの具体的なことについては４章で示す．

また，未登録語に対処する目的で片仮名・アルファベット列や１文字の漢字，平仮名を単語と同等に扱うために，コスト最小法だけを用いた表方式のアルゴリズムでは入力文の１文字毎に自立語辞書の検索を行う可能性がある．それを避けるために，本アルゴリズムでは文節末の可能性の強さに関するヒューリスティックを用いて解析表の展開を多段階に制御している．次に，文節末の可能性の強さを値とする関数bendを定義する．この定義では文節末の可能性の強さをδ段階に分類し，値が大きいほど文節末の可能性が強いとする．

［定義４］　関数bend

文法情報αに対して，もしend(α)でないならば，bend(α)=0，end(α)ならば，文法情報αおよびその他の条件（入力文における字種情報等）に応じて，bend(α)=1,2,…,δとする．　■

関数bendの具体的な定義についても４章で示す．

３．３　アルゴリズム

入力文字列における位置を表す整数i，j，文法情報α，文節末の可能性の強さを表す値e，コストの部分和w，0か1の２値を値とする整数cからなる６項組(i,j,α,e,w,c)を項目と呼び，項目を要素とする集合T(0),T(1),…,T(n)の全体を解析表と呼ぶ．また，c=0である項目を非活動中の項目と呼び，c=1である項目を活動中の項目と呼ぶ．文節数最小法やコスト最小法を用いたアルゴリズムでは，解析表を作成する時点で不要と判定された項目は捨てていたが，このアルゴリズムでは，解析表を狭く展開する時点で不要であると判定した項目でも，後戻りが起こって，より広い解析表を展開する時点で必要になる可能性がある．そのときに単語辞書を重複して検索することを避けるために，不要な項目は非活動中の状態にして登録する．

解析表の作成は，入力文字列を文頭から文末に向かって走査する過程で，単語に対応する項目を作り適当な集合Tに登録することにより行われる．この過程で作成された活動中の項目におけるjの最大値を変数Rmaxで記憶する．

Ctable(0),Ctable(1),…,Ctable(n-1)
Ftable(1),Ftable(2),…,Ftable(n-1)

は共に値として0か1をとる配列で，Ctable(i)の値は入力文字列における位置iで自立語辞書の検索を行ったか否かを，Ftable(i)の値は同じく付属語辞書の検索を行ったか否かを示している．また，

Etable(1),Etable(2),…,Etable(n)

は値として0,1,…,δをとる配列で，Etable(i)の値は入力文字列における位置iにおける文節末の可能性の強さの最大値を示している．本稿で示すアルゴリズムでは，Etableの値に従って解析表の展開をδ段階に制御する．解析表を作成している過程で，どの段階にあるかを変数Phaseで表す．

次にアルゴリズムの記述に用いる幾つかの基本的な手続きを定義する．

解析表に項目を登録する手続きtadd(項目)を次のように定義する．

[手続き　tadd]
与えられた項目(i,j,α,e,w,c)に対して，
c=1ならば，
　項目(i,j,α,e,w,c)をT(j)に登録する．
　j>RmaxならばRmax←j．
　e>Etable(j)ならばEtable(j)←e．
　e>PhaseならばPhase←e．
c＝０ならば，
　項目(i,j,α,e,w,c)をT(i)に登録する．　■

逆にT(i)から項目(i,j,α,e,w,0)を削除する手続きをtdelete(項目)とする．

次に入力文字列における位置jから始まる自立語の項目を作成して解析表に登録する手続きcmake(j)を定義する．

[手続き　cmake]
Ctable(j)=0ならば，
　Cset(j)を求める．
　T(j)中のe>Phaseであるすべての項目(i,j,β,e,v,1)におけるvの最小値を求め，それをwとする．
　Cset(j)のすべての要素(j,k,α)について，
　　tadd((j,k,α,bend(α),w+cost(α),1))．
　Ctable(i)←1.　■

また，入力文字列における位置jから始まる付属語の項目を作成して解析表に登録する手続きfmake(j)を次のように定義する．

[手続き　fmake]
Ftable(j)=0ならば，
　Fset(j)を求める．
　Fset(j)のすべての要素(j,k,β)について，
　　T(j)にconnect(α,β)を満たす項目(i,j,α,e,v,1)が存在するならば，
　　　そのようなvの最小値wについて，
　　　tadd((j,k,β,bend(β),cost(β)+w,1))．
　　存在しないならば，
　　　tadd((j,k,β,bend(β),0,0))．
Ftable(j)=1ならば，
　T(j)の要素(j,k,β,bend(β),0,0)すべてについて，
　　T(j)にconnect(α,β)を満たす項目(i,j,α,g,v,1)が存在するならば，そのようなｖの最小値wについて，
　　　tdelete((j,k,β,bend(β),0,0))．
　　　tadd((j,k,β,bend(β),cost(β)+w,1))．■

これらの手続きを用いて解析表を作成するアルゴリズムは次のようになる．

[解析表作成アルゴリズム]

［０］　初期設定

Rmax←0.

Phase←δ.

Etable(i)←0　　(i=1,2,…,n).

Ctable(i)←0　　(i=0,1,…,n-1).

Ftable(i)←0　　(i=1,2,…,n-1).

T(i)←ε　　(i=0,1,…,n).

p←1.

［１］　文頭の自立語の検索

cmake(0).

［２］　位置pからの単語の検索

fmake(p).

Etable(p)≧Phaseならばcmake(p).

［３］　ポインタの設定

p←p+1

p=nならば終了.

p>Rmaxならば，

Phase←Phase-1.

Etable(p)=δまたはp=1になるまで，

p←p-1.

［２］に行く.　■

項目が解析表に属することについては，次の定理が成り立つ．

［定理１］　項目の意味

解析表を作成している過程でT(k)に項目(i,j,α,bend(α),w,c)が存在することは次のことを意味する．

(1)　c=0のとき．

i=kかつw=0であり，入力文字列の部分列s(i+1)s(i+2)…s(j)は文法情報αをもつ単語であるが，文節の文法モデルを満足する文頭からの単語の連鎖が解析表中に存在しない．

(2)　c=1のとき．

j=kかつ入力文字列の部分列s(i+1)s(i+2)…s(j)は文法情報αをもつ単語であり，文節の文法モデルを満足する文頭からの単語の連鎖が解析表中に存在し，そのような連鎖のコストの総和の最小値はwである．

■

証明はアルゴリズムより明らかであるので省略する．

アルゴリズムにおいて，条件p>Rmaxが成り立ったときの処理が後戻りの処理である．ここでは，二つ以上前にある文節末の可能性が最も強い位置（Etable(p)=δの位置）まで後戻りしても解析の精度は殆ど改善されないという実験結果から，一つ前にある文節末の可能性が最も強い位置まで後戻りしている．

このアルゴリズムを実行した結果T(n)に状態が１で文節末の可能性が０でない項目が存在するならば，入力文の形態素解析の結果を出力することができる．解析結果を抽出するためには，まず上記の条件を満たす項目でコストの総和（項目の第５項目）が最小であるものを選び，以後解析表を文頭側に向かって述語connectを満たし，状態が１である項目の連鎖を取り出せばよい．解析結果抽出のアルゴリズムは明らかなため省略する．

解析表作成アルゴリズムの能率については次の定理が成り立つ．

［定理２］　解析表作成アルゴリズムの能率

解析表作成アルゴリズムの能率は，入力文字列の長さnに対して時間計算量，領域計算量共にO(n)である．

［証明］　集合T(j)に属する項目の個数を考える．集合T(j)には次の２種類の項目が存在する．

(i,j,α,bend(α),w,1)

(j,k,β,bend(β),0,0)

単語の長さおよび一つの文字列がなり得る単語の個数（同形異義語の個数）は有限であるから，これら２種類の項目においてjを固定した場合，i，αおよびk，βが取り得る値の個数はある定数でおさえることができる．また，一つのi，j，αの組み合わせに対してwの値は一通りである．したがって，集合T(j)に属する項目の個数はO(1)であるから，解析表全体の大きさはO(n)である．配列Etable，Ctable，Ftableの大きさもそれぞれO(n)であるから，解析表作成アルゴリズムの領域計算量はO(n)である．

前半の議論と同様にして集合Cset(j)，Fset(j)，T(j)に属する項目の個数はO(1)であるから，手続きcmake，fmakeの１回の実行に要する時間計算量はO(1)である．解析表作成アルゴリズムにおいて，ステップ［２］，［３］の実行は高々δ×n回であ

る．したがって，解析表作成アルゴリズムの時間計算量はO(n)である． ■

4．実験

4．1　実験の手順

実験の手順を図２に示す．まず，解析表作成プログラムで入力文を構成する可能性があるすべての単語の項目を作成し，正しい解析結果を構成する項目に人手で印を付けてファイルに格納する．コスト設定用プログラム，解析の精度および能率の評価用プログラムは，この解析表のファイルを入力データとして処理を行う．実験システムは主記憶2MBのVAX11/750 VAXVMS上にFORTRANで実現した．実験システムの単語辞書には約30,000語の自立語辞書と約4,000語の付属語辞書を用いた．

4．2　コストの設定

解析表を構成する項目のコストは，自立語，付属語，活用語尾，片仮名・アルファベット列，一文字漢字，一文字平仮名，記号（句読点）の分類に対して設定する．これらのコストの値は，すべてのコストの初期値を０としておき，繰り返し修正法を用いて決定した．それぞれのコストをC_i(i=1,2,…,7)とすると，一つのコストの修正方法は，

$C_i \leftarrow C_i+1$

$C_i \leftarrow C_i$　（修正なし）

if $C_i>0$ then $C_i \leftarrow C_i-1$

の３通りとし，一文の入力データに対して一つのコストを修正する．修正の対象とするコストは，そのコストの修正後に，コストの総和が小さい解析結果から順番に出力したとき正しい解析結果の出力される順番が最も早いものを選択する．このようなコストが複数個存在する場合には，そのときに作成した項目の総数が少ない方を選択する．ただし，一般にコストの総和が等しい解析結果が複数個存在するため，正しい解析結果が出力される順番を一意に決定することはできない．今回の実験では，コストの総和が等しい複数個の解析結果の中から一つの解析結果を取り出す確率は等確率であると仮定し，コストの総和がi番目に小さい解析結果の数をm_i，正しい解析結果のコストの総和がj番目に小さいときの正しい解析結果が出力される順番の期待値を次式で求めた．

$$(m_j+1)/2 + \sum_{i=1}^{j-1} m_i$$

また，繰り返し修正の過程で，正しい解析結果の出力される順番が10,000以上になる入力データは評価の対象から除外した．

コスト設定用のデータとしては，科学技術雑誌の一論文を構成する662個の句読点で終わる文字列（総文字数14,275）を用いた．このデータを繰り返し用いてコストの繰り返し修正を行った結果，４回目にコストの変化が定常状態に入った． コストの変化が定常状態になったときの各コストの662文に対

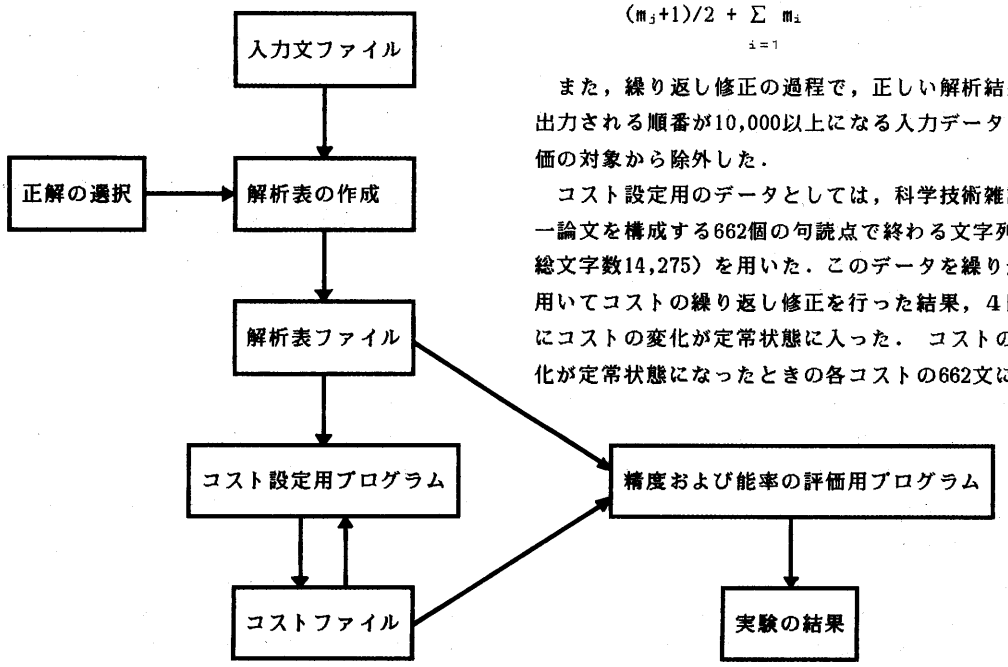


図２　実験の手順

する平均値を表１に示す．ここで，記号のコストは他に競合するものがないので０に固定している．実験では表１のコストを整数化して得られる表２のコストを用いた．

４．３　文節末の可能性の強さ

解析表の展開を制御するために用いる文節末の可能性の強さは表３に示す８段階に分類する．したがって，解析表の展開は７段階に制御されることになる．表３において字種変化とは，注目している単語の末尾の文字が平仮名で，その文末側の単語の先頭の文字が非平仮名であることを示している．また，自立語，付属語，活用語尾における文節末の可能性の強弱は，品詞と活用形で決定している．

４．４　解析精度と能率の比較

解析表の展開を制御するアルゴリズムと制御しないアルゴリズムに対する解析精度と能率の比較実験は，コストの設定に用いた662個の文字列と，同じく科学技術雑誌から取り出した別の１論文を構成する692個の句読点で終わる文字列（総文字数15,587）を用いて行った．以下では，前者をテキスト１，後者をテキスト２とよび，解析表の展開を制御しない方式を旧方式，制御する方式を新方式とよぶ．

実験の結果を表４に示す．表４において，Emin，Emaxの欄はコストの総和が最小となる解析結果における誤り率の最小値と最大値の平均である．ここで誤り率は次式によって求めた．

$$\text{誤り率} = \frac{\text{解析を誤った部分の入力文における長さ}}{\text{入力文の長さ}}$$

解析結果の個数の欄には，同じくコストの総和が最小となる解析結果の個数の平均値を示している．また，自立語辞書の検索回数および付属語辞書の検索回数の欄には，一つの文字列の解析中に行った自立語辞書と付属語辞書の検索回数の平均値を示している．

なお，コストの総和が最小となる解析結果の個数が20,000個以上になって評価の対象から除外した文字列の個数は，テキスト１に対して旧方式では３個，新方式では４個，テキスト２に対しては両方式とも１個であった．また，コストの総和が最小となる解析結果の中に正しい解析を含んでいたものは，テキスト１に対して旧方式では477個，新方式では438

表１　変化が定常状態になったコストの平均値

種類	コスト値
自立語	２．１５
付属語	０．５７
活用語尾	０．５２
片仮名・アルファベット列	３．３５
一文字漢字	１２．４
一文字平仮名	２３．９
記号	０．０

表２　実験で使用したコストの値

種類	コスト値
自立語	４
付属語	１
活用語尾	１
片仮名・アルファベット列	７
一文字漢字	２５
一文字平仮名	４８
記号	０

表３　文節末の可能性の強さ

	文節末	字種変化	強さ
自立語，付属語，活用語尾	可（強）	有	７
	可（強）	無	６
	可（弱）	有	５
	可（弱）	無	４
	不可	－	０
片仮名・アルファベット列		－	３
一文字漢字		－	２
一文字平仮名		－	１

表４　実験結果

テキスト１	Emin(%)	Emax(%)	解析結果の個数（個／文）	自立語辞書の検索回数（回／文）	付属語辞書の検索回数（回／文）
新方式	４．９９	２０．４	１２５	８．２	１９．６
旧方式	４．４６	２０．５	１３９	２７．３	２５．６

テキスト２	Emin(%)	Emax(%)	解析結果の個数（個／文）	自立語辞書の検索回数（回／文）	付属語辞書の検索回数（回／文）
新方式	５．６０	２０．５	１６８	８．９	２０．６
旧方式	４．３８	１９．９	１７４	２８．３	２６．９

個，テキスト２に対して旧方式では479個，新方式では427個であった．

４．５　実験結果の考察

まず，コストの総和が最小となる解析結果のあいまいさが多いが，その原因は今回の実験に用いた付属語辞書における単語のカテゴリーが構文的な情報だけでなく意味的な情報も区別して分類されており，その数も含めて計数しているためである．例えば，格助詞「に」は４通りのあいまいさを持っている．

次に，コスト最小法において解析の精度を悪くしている原因は，付属語辞書の不備（＜動詞・連用形＞＋かねない）や長単位の付属語的表現が本質的にもつあいまいさ（を用いている）である．

表４からも明らかなように，表の展開を制御するアルゴリズムは能率の点では改善されているが，精度は悪くなっている．その主な原因としては，字種情報を利用しているために平仮名書きの自立語が存在すると，狭く展開した表の中に正解が入らない可能性があるということが考えられる．これに対しては頻繁に平仮名書きされる自立語の辞書を用意する必要がある．

５．あとがき

解析表の展開を制御する表方式の形態素解析アルゴリズムとその精度および能率に関する実験について述べた．今後の課題としては，解析精度の改善，接辞処理，数詞処理の強化等が考えられる．

最後に，本実験で使用した自立語辞書は「九州芸工大自立語辞書KID-82」(6)の規模を９万語から３万語に筆者らが縮小したものである．KID-82の使用を認めて頂いた九州芸工大・稲永紘之講師および本研究に対して貴重な御助言を頂いた九州大学・日高達助教授に感謝の意を表する．

参考文献

(1) 吉村・日高・吉田：未登録語を含む日本語文の形態素解析アルゴリズム，九州大学工学集報，Vol.55,No.6,1982.

(2) 吉村・日高・吉田：文節数最小法を用いたべた書き日本語文の形態素解析，情報処理学会論文誌，Vol.23,No.6,1982.

(3) 吉村・日高・吉田：漢字仮名混り文の形態素解析とその実験，情報処理学会第28回全国大会講演論文集，5M-6,1984.

(4) 吉村・渡辺・津田・首藤：広がりを制御できる表方式の形態素解析アルゴリズム，情報処理学会第32回全国大会講演論文集，2T-5,1986.

(5) Knuth,D.E.:The Art of Computer Programming, Vol.3, Sorting and Searching, Addison-Wesley Pub. Co.,1973.

(6) 稲永・吉田：日本語処理のための機械辞書，情報処理，Vol.23.No.2,1982.