未登録語を含む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††

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文法情報等の推定は不可欠な処理である．日本語文の解析手順は，形態素解析，構文解析，意味解析などの各解析を段階的に行うものと，これらを融合的に行うものとに大きく分類できる．本論文では前者の方式を想定し，形態素解析の段階における未登録語の処理について述べる．本論文で示す形態素解析アルゴリズムは基本的に解析表を利用した横型探索のアルゴリズムであり，入力文中の一文字の漢字，平仮名や英字列，片仮名列を自立語と同等に扱うことにより未登録語の処理を可能にしている．このとき入力文の一文字ごとに自立語辞書を検索するという効率の問題やシステムにとっては正しいが本質的には誤っている膨大な数の解析が発生するという尤度評価の問題が生じる．これに対して本アルゴリズムでは，字種情報に基づいた文節末の可能性と解析の単位に対するコストの付与という二つのヒューリスティック情報を利用している．アルゴリズムの能率は入力文の文字数 n に対して時間計算量，領域計算量ともに $O(n)$ である．また，このアルゴリズムにより入力文中の未登録語の 90.9% を正しく処理できることを実験により確認した．

1. まえがき

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（システムの辞書に登録されていない単語）の位置や文法情報等の推定は不可欠な処理である．日本語文の解析手順は，形態素解析，構文解析，意味解析などの各解析を段階的に行うものと，これらを融合的に行うものとに大きく分類できる．本論文では前者の段階的な解析手順を想定し，漢字仮名混じり日本語文の形態素解析アルゴリズムとその未登録語処理について述べる．

一般に漢字仮名混じり日本語文の形態素解析には，字種による文節等への強制分割，単語辞書の検索，品詞や活用形等に基づいた単語間の接続検査の三つの処理を利用するもの[1)~3)]と，統計的手法に基づいたもの[4),5)]がある．前者における未登録語の処理としては，縦型探索において単語の抽出に失敗した時点で，左からの最短文字列を未登録語として処理を進めるもの[6)]や字種情報を用いて文字を読み飛ばして処理を進めるもの[7)]などが報告されているが，それらの方法の精度については報告されていない．また後者は未登録語の処理を取り入れやすい枠組ではあるが[8),9)]，この種の処理の具体的な報告はない．

筆者らは，上記の単語辞書の検索と品詞や活用形等に基づいた単語間の接続検査の二つの処理に，ヒューリスティックとして文節数最小法[10)]の考え方を一般化したコスト最小法を組み合わせて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を提案した[11)]．このアルゴリズムでは，システムの単語辞書に登録されていない片仮名列，アルファベット列や一文字の漢字，平仮名を自立語と同等に扱うことで未登録語に対処している．しかし，一文字の漢字，平仮名を自立語と同等に扱うために，入力文中の一文字ごとに自立語辞書の検索を行うという欠点を持っている．本論文では，字種情報等から決定する文節末の可能性の強さに関するヒューリスティックを利用して不必要な自立語辞書の検索を行わないように改良したアルゴリズムとその能率，精度を評価するために行った実験について報告する．このアルゴリズムの特徴としては，横型探索である表方式に表の展開を制御する機構を取り入れて，もっともらしい解析結果から出力できる横型探索の長所とヒューリスティックが正しく機能した場合に正解が短時間で得られる縦型探索の長所を合わせ持つことである．

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を仮定する．

（1）〔文〕::=〔文節〕|〔文〕・〔文節〕

（2）〔文節〕::=〔自立語〕|〔文節〕・〔付属語〕

本論文で示すアルゴリズムでは，（2）に示す文節の構

† Morphological Analysis of Japanese Sentences Containing Unknown Words by KENJI YOSHIMURA, MITSUNO TAKEUCHI, KENZO TSUDA and KOSHO SHUDO (Department of Electronics, Faculty of Engineering, Fukuoka University).

†† 福岡大学工学部電子工学科

造を規定する規則のみを用いて，(1)に示す文の構造を規定する規則は用いていない．なお，実験システムでは〔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[12]を用いているが，アルゴリズムに関する議論においてこ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 節の (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，$s=s(1)s(2)\cdots s(n)$ とする．

[定義1] 諸述語の定義

(1) 入力文字列の部分列 $s(i+1)s(i+2)\cdots s(j)$ 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 W1 と文法情報 β の単語 W2 の連接 W1W2 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 の単語が文節の末尾になれることを 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を次のように定義する．ただし，このモデルは文節を構成する単語の並びを隣接する二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．

[定義2] 文節の文法モデル

文字列 $s=s(1)s(2)\cdots s(n)$ 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) $(j=0,1,\cdots,m)$

(2) cword(α_0)

(3) connect(α_j, α_{j+1}) $(j=0,1,\cdots,m-1)$

(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． ■

2.3 未登録語

本論文で示すアルゴリズムでは，システムの単語辞書に登録されていない片仮名列，アルファベット列や一文字の漢字，平仮名を自立語と同等に扱うことで未登録語に対処する．このとき，次のように定義されるWに対応する文字列を未登録語として認定する．

X::=[一文字漢字]|[一文字平仮名]
|[片仮名列]|[アルファベット列]
N::=[名詞]
G::=[語尾]
X::=XX|XN|NX|NN
W::=X|XG

ただし，促音(っ)とよう音（ゃ，ゅ，ょ）は直前の文字と一緒にして［一文字平仮名］とする．また，この定義では名詞列として解析された未登録語と本来の名詞列を区別できないため，構文解析を行う前にこれらを識別する名詞列の処理が必要である．

3. 形態素解析アルゴリズム

3.1 アルゴリズムを記述するための準備

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し，この解析表を文末側から文頭側に走査して解析結果を出力する．解析表から解析結果を取り出すアルゴリズムの構成は，解析表を作成するアルゴリズムの内容から明らかなので，本論文では解析表を作成するアルゴリズムだけを示す．

次にアルゴリズムを簡潔に記述するために集合 Cset と Fset 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$(i)=\{(i,j,\alpha)|\text{word}(i,j,\alpha)\ \&\ \text{cword}(\alpha)\}$

Fset$(i)=\{(i,j,\alpha)|\text{word}(i,j,\alpha)\ \&\ \text{fword}(\alpha)\}$ ■

ここで，Cset(i) は $s(i+1)$ の字種に応じて，[一文字漢字]，[一文字平仮名]，[片仮名列] または [アルファベット列] に対応する要素も含むものとする．

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 $s(i+1)s(i+2)\cdots s(n)$ 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．この手続きの時間計算量は，単語辞書のデータ構造として TRIE 構造[13]を用いることにより $O(1)$ になる．

3.2 ヒューリスティック

本論文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．

単語の連鎖を2単語間の連接規則だけで規定した文節の文法モデルだけを用いて複合語の分割や未登録語の処理を行うと，正しい解析結果の他に多くの誤った

解析結果が発生する．これらの解析結果の中から正しい解析結果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[14]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 $\mathrm{cost}(\alpha)$ とする．関数 cost の具体的な定義は4章で示す．

また，コスト最小法を用いた形態素解析アルゴリズムでは，未登録語に対処するために片仮名列，アルファベット列や一文字の漢字，平仮名を単語と同等に扱っているために，入力文の一文字ごとに自立語辞書の検索を行う可能性がある．不必要な自立語辞書の検索をできるだけ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いる．文法情報 α に対して，その文節末の可能性の強さを与える関数を $\mathrm{bend}(\alpha)$ とする．文節末の可能性の強さを $\delta+1$ 段階に設定した場合，$\mathrm{bend}(\alpha)$ は0から δ までの整数値をとり，その値が大きいほど文節末の可能性が強いとする．関数 bend の具体的な定義も4章で示す．

3.3 アルゴリズム

3.3.1 変数の説明

入力文字列における単語の先頭と末尾の位置を示す整数 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値をとる整数 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 $T(0)$, $T(1), \cdots, T(n)$ 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，$c=1$ である項目を活動中の項目と呼ぶ．解析表は，入力文字列を文頭から文末に向かって走査し，単語に対応する項目を作り，単語の末尾の位置 j に従って集合 $T(j)$ に登録して作成する．

解析表を作成する過程で登録された活動中の項目における単語の末尾を示す整数 j の最大値を変数 Rmax で記憶する．配列 $\mathrm{Ctable}(0:n-1)$, $\mathrm{Ftable}(1:n-1)$ は共に値として0か1をとる配列で，$\mathrm{Ctable}(i)$ の値が1であること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ことを示している．同様に $\mathrm{Ftable}(i)$ の値は付属語辞書の検索状況を示している．また，配列 $\mathrm{Etable}(1:n)$ は値として0，1，$\cdots$，δ をとり，入力文字列の位置 i における文節末の可能性の強さの中で最大のものを示している．このアルゴリズムでは，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．

これらの変数は以下で定義する手続きにおいて大域的である．

3.3.2 基本的な手続き

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．

$j>\mathrm{Rmax}$ ならば $\mathrm{Rmax}\leftarrow j$.

$e>\mathrm{Etable}(j)$ ならば $\mathrm{Etable}(j)\leftarrow e$.

$e=\delta$ ならば $\mathrm{Phase}\leftarrow\delta$.

$c=0$ 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 $(i, j, \alpha, e, w, 0)$ 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 $\mathrm{cmake}(j)$ を定義する．

[手続き cmake]

$\mathrm{Ctable}(j)=0$ ならば，

$\mathrm{Cset}(j)$ を求める．

$T(j)$ 中の $e>\mathrm{Phase}$ であるすべての項目 $(i, j, \beta, e, v, 1)$ における v の最小値を求め，それを w とする．$\mathrm{Cset}(j)$ のすべての要素 (j, k, α) について，

$\mathrm{tadd}((j, k, \alpha, \mathrm{bend}(\alpha), w+\mathrm{cost}(\alpha), 1))$.

$\mathrm{Ctable}(i)\leftarrow 1$. ■

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 $\mathrm{fmake}(j)$ を次のように定義する．

[手続き fmake]

$\mathrm{Ftable}(j)=0$ ならば，

$\mathrm{Fset}(j)$ を求める．

$\mathrm{Fset}(j)$ のすべての要素 (j, k, β) について，

$T(j)$ に $\mathrm{connect}(\alpha, \beta)$ を満たす項目 $(i, j, \alpha, e, v, 1)$ が存在するならば，

そのような v の最小値 w について，

$\mathrm{tadd}((j, k, \beta, \mathrm{bend}(\beta), w+\mathrm{cost}(\beta), 1))$.

存在しないならば，

$\mathrm{tadd}((j, k, \beta, \mathrm{bend}(\beta), 0, 0))$.

$\mathrm{Ftable}(j)=1$ ならば，

$T(j)$ のすべての要素 $(j, k, \beta, \mathrm{bend}(\beta), 0, 0)$ につ

いて，

$T(j)$ に connect(α, β) を満たす項目 $(i, j, \alpha, g, v, 1)$ が存在するならば，そのような v の最小値 w について，

tdelete$((j, k, \beta, \text{bend}(\beta), 0, 0))$.

tadd$((j, k, \beta, \text{bend}(\beta), w+\text{cost}(\beta), 1))$. ■

3.3.3 解析表作成アルゴリズム

解析表を作成するアルゴリズムは，以上の手続きを用いて次のように表される．なお，アルゴリズムの記述中で ε は空集合を表している．

[解析表作成アルゴリズム]

[0] 初期設定

Rmax←0.

Phase←δ.

Etable(i)←0 $(i=1, 2, \cdots, n)$.

Ctable(i)←0 $(i=0, 1, \cdots, n-1)$.

Ftable(i)←0 $(i=1, 2, \cdots, n-1)$.

$T(i)=\varepsilon$ $(i=0, 1, \cdots, n)$.

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

$T(p)\neq\varepsilon$ ならば，

fmake(p).

Etable$(p)\geqq$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$(p)=\delta$ または $p=1$ になるまで，

p←p−1.

[2]に行く. ■

このアルゴリズムは，文節末の可能性の強さに関するヒューリスティックで全体の流れを制御しているために，コストの総和が最小となる解析結果が得られることは保証していない．この点を明確にするために，このアルゴリズムで作成した解析表においてコストの総和が最小であることをコストの総和が極小であるといい，そのような解析結果を極小解と呼ぶ．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 $(i, j, \alpha, \text{bend}(\alpha), w, c)$ 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 $s(i+1)s(i+2)\cdots s(j)$ は文法情報 α を持つ単語であるが，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 $s(i+1)s(i+2)\cdots s(j)$ 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，その連鎖のコストの総和は 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．

アルゴリズムの[3]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$(p)=\delta$ の位置) まで後戻りしても解析の精度はほとん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 $c=1$ 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．解析結果を取り出すためには，まず上記の条件を満たす項目でコストの総和 w 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文節の文法モデルを満たし，$c=1$ である項目の連鎖を取り出せばよい．

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] $T(j)$ に属する項目の個数を考える．$T(j)$ には $(i, j, \alpha, \text{bend}(\alpha), w, 1)$ と $(j, k, \beta, \text{bend}(\beta), 0, 0)$ の二種類の項目が存在する．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，これら二種類の項目において j を固定した場合，i, k および α, β が取り得る値の場合の数はある定数でおさえることができる．また，手続き cmake, fmake の定義より一つの i, j, α または j, k, β の組合せに対して w の値は1通りである．したがって，$T(j)$ 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．同様にして集合 Cset(j), Fset(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 δn 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． ■

このアルゴリズムにおいて，変数 Phase の値を1に固定すると完全な解析表を作成する従来の表方式のアルゴリズムとなる．以下では，上に示した［解析表作成アルゴリズム］をアルゴリズム T+と呼び，アルゴリズム T+ において Phase の値を1に固定した従来の表方式のアルゴリズムをアルゴリズムTと呼ぶ．アルゴリズムTでは，コストの総和が最小である解析結果が極小解となる．

4. 実　　験

4.1 実験システム

実験には解析の対象とする日本語文として三つの科学技術記事[15)~17)]の本文を構成するそれぞれ 662 個（総文字数 14,275），692 個（総文字数 15,587），545 個（総文字数 10,320）の句読点で終る文字列を使用した．この入力文データに対しては，数式および括弧を用いた文中の注釈を取り除くこと以外の修正は行っていない．以下では，この三つの入力文データをそれぞれテキスト1，テキスト2，テキスト3と呼ぶ．

実験システムは VAX 11/750 VAXVMS 上に FORTRAN 77 で作成した．システムの単語辞書としては，「九州芸工大自立語辞書 KID-82」[18)] の規模を9万語から基本語の3万語に縮小して副詞，接続詞，連体詞などの慣用的表現を追加した自立語辞書と約4千語の付属語辞書[19)]を使用した．

4.2 コスト値

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）の七種類の分類に対して設定している．実験で使用したコスト値を**表1**に示す．この値はテキスト1に対して繰り返し修正法を用いて決定したものである[14)]．

4.3 文節末の可能性の強さ

文節末の可能性の強さは**表2**に示す8段階（$\delta=7$）に分類した．表2において，種類はコストの設定を行った分類に対応する．文節末の可能性の強さは，辞書に登録されている単語で文節末に成り得るものには大きな値を与え，未登録語に対処するためのものには小さな値を与える．さらに，前者の値は字種変化と文法情報を利用して4段階（7，6，5，4）に細分している．字種変化の有無とは注目している単語の末尾の文字が平仮名で，その右側の文字が平仮名以外の文字か否かを示している．また，文節末の可，不可および可の場合の強弱の表示は単語の品詞，活用形から決定している．例えば，活用しない単語では，名詞は可/弱でその他のものは可/強となり，活用語尾では音便変化していない連用形で可/弱，終止形，連体形，命令形で可/強，その他で不可となる．

4.4 解析の精度と能率の比較

ここでは，テキスト1，テキスト2，テキスト3の1,899 文を用いて行ったアルゴリズムTとアルゴリズ

表1 実験で使用したコスト
Table 1 The costs used in the experiments.

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2 文節末の可能性の強さ
Table 2 The values indicate possibility to terminate a BUNSETSU.

種　　類	文節末	字種変化	強さ
自立語	可／強	有	7
自立語	可／強	無	6
付属語	可／弱	有	5
活用語尾	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3 精度と能率の比較（1）
Table 3 Comparison of accuracy and efficiency (1).

	E_{min}(%)	E_{max}(%)	コストが極小である解析結果の個数	自立語辞書の検索回数	付属語辞書の検索回数
アルゴリズムT	4.67	20.7	139	26.9	25.4
アルゴリズムT+	4.59	20.2	138	8.30	19.8

ム T+ の精度と能率の比較実験の結果について述べる．

実験の結果を表 3 に示す．表3において，E_{min}, E_{max} の欄は極小解の誤り率の最小値と最大値の平均である．ここで誤り率とは，解析を誤った部分の入力文における長さを入力文の長さで割った商である．解析結果の個数の欄には極小解の個数の平均値を示している．次に，アルゴリズムの能率は単語辞書の検索回数で比較している．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，一つの入力文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．ここで，極小解の個数が多いのは，今回の実験で用いた付属語辞書の品詞分類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いるためである．例えば，一つの格助詞「に」があるだけで4通りのあいまいな解析結果が生じる．

なお，今回の実験では極小解の個数が2万個以上になる文は評価の対象から除外した．それぞれのアルゴリズムにおいて，除外した入力文の個数と正しい解析が極小解になった入力文の個数を表 4 に示す．

アルゴリズムTにおいて，極小解の中に正しい解析が含まれなかった原因の主なものとその割合を次に示す．例において○印は正しい解析を示し，×印はコストが最小となる解析を示している．

（1） 自立語を付属語的表現の一部として解析する (45.9%)．

○ ～ を [付属語] 用い [動詞＋語尾] ている [付属語]．

× ～ を用いて [付属語] いる [動詞＋語尾 (居る)]．

（2） 活用する語とその活用語尾を名詞，連体詞等として解析する (18.2%)．

○ ～ が [付属語] あり [動詞＋語尾 (「有り」)]，

× ～ が [付属語] あり [名詞 (「蟻」)]，

（3） 未登録語がある (14.6%)．

○ ～ 入 [漢字] 射 [漢字] 電子 [名詞] 量 [名詞]

× ～ 入 [漢字] 射 [動詞 (「射る」の連用形)] 電子 [名詞] 量 [名詞]

（4） 付属語列を一つの付属語的表現として解析する (11.2%)．

○ ～ と [格助詞] は [係助詞] ・ 独立して

× ～ とは [付属語 (「というのは」の意)] 独立して

アルゴリズム T+ において極小解の中に正しい解析が含まれなかった原因もほぼ同様であるが，アルゴリズムTにおいて付属語列を自立語として解析するために正しい解析結果が極小解にならなかったものが改善されている．

○ 先に [副詞] 述べ [動詞] た [助動詞] ように [付属語]

× 先に述べた [連体詞] ように [副詞 (「陽に」)]

その反面，アルゴリズムTにおいて正しい解析結果が極小解となっていた平仮名表記の自立語に関してはコストが極小にならなくなったものもある．

○ 全部 [名詞] もしくは [接続詞] 一部 [名詞]

× 全部 [名詞] も [助詞] しく [動詞] は [名詞] 一部 [名詞]

4.5 未登録語の処理

アルゴリズムTとアルゴリズム T+ における未登録語処理の精度をそれぞれ表 5 と表 6 に示す．表に

表 4 精度と能率の比較 (2)
Table 4 Comparison of accuracy and efficiency (2).

	入力文の総数	評価できなかった文数	コストが極小となる解析に正解があった文数
アルゴリズムT	1,899	4	1,343 (70.7%)
アルゴリズムT+	1,899	4	1,316 (69.3%)

表 5 未登録語の解析 (アルゴリズムT)
Table 5 Analysis of the unknown words (algorithm T).

		名詞	サ変名詞	動詞	形容動詞	その他	計
正しく解析できたもの	名詞列	899	8				907
	未登録語	1,159	17	1	15		1,192
誤って解析したもの	名詞列	18					18
	未登録語	39	12	1	24	5	81

表 6 未登録語の解析 (アルゴリズム T+)
Table 6 Analysis of the unknown words (algorithm T+).

		名詞	サ変名詞	動詞	形容動詞	その他	計
正しく解析できたもの	名詞列	886	8				894
	未登録語	1,122	17	1	17		1,157
誤って解析したもの	名詞列	31					31
	未登録語	76	12	1	22	5	116

おいて名詞列とは未登録語ではない名詞の列を意味しており，未登録語とは一つの未登録語または未登録語を含む名詞の列を意味している．アルゴリズム T+では自立語辞書の検索位置を制限しているために極小解の中に正しい解析が含まれなくなったものがある．

○ トランスファー［片仮名列］ 方式［名詞］
× トランスファー［片仮名列］ 方［接尾語］式［名詞］

表5と表6より純粋な未登録語の処理の精度はアルゴリズムTで 93.6%，アルゴリズム T+で 90.9% である．

5. あとがき

本論文では日本語文の形態素解析アルゴリズムとその実験結果について述べた．アルゴリズム T+ はアルゴリズムTに比べて未登録語の解析精度は若干悪くなっているが処理能率に関しては3倍程度の改善が得られた．なお，4章で示したように今回の実験では長単位の付属語的表現が解析精度を悪くする原因の半数近くになっている．長単位の付属語的表現は構文・意味解析にとっては有意義であるが，本方式の形態素解析を採用した場合には短単位の付属語を使用し，構文解析の前処理として長単位の付属語的表現を認定する方法が適していると考えられる．

謝辞 KID-82 の使用を認めていただいた九州芸工大・稲永紘之講師および日頃貴重な助言をいただく九州工業大学・吉田 将教授，九州大学・日高 達教授に感謝の意を表する．なお本研究の一部は文部省科研費「特定研究(1)言語情報処理の高度化」によった．

参考文献

1) 長尾，辻井，山上，建部：国語辞書の記憶と日本語文の自動分割，情報処理，Vol. 19, No. 6, pp. 514-521 (1978).
2) 中野，野村：日本語の形態素分析，情報処理，Vol. 20, No. 10, pp. 857-864 (1979).
3) 高橋編：日本語情報処理，pp. 64-74，近代科学社，東京 (1986).
4) 藤崎：動的計画法による漢字仮名混じり文の単位切りと仮名ふり，情報処理学会自然言語処理研究会資料，28-5 (1981).
5) 松延，日高，吉田：確率文節文法による構文解析，情報処理学会自然言語処理研究会資料，56-3 (1986).
6) 電子技術総合研究所推論機構研究室：拡張 LINGOL，p. 9 (1978).
7) 坂本：日本語形態素解析の基本設計，情報処理学会自然言語処理研究会資料，38-3 (1983).
8) 福永，松延，日高，吉田：漢字と読みの組を造語単位とした単語の造語モデル，情報処理学会自然言語処理研究会資料，59-3 (1987).
9) 武田，藤崎：統計的手法による漢字複合語の自動分割，情報処理学会論文誌，Vol. 28, No. 9, pp. 952-961 (1987).
10) 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, pp. 40-46 (1983).
11) 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, pp. 635-639 (1982).
12) 首藤，楢原，吉田：日本語文の機械処理のための文節構造モデル，電子通信学会論文誌，Vol. J62-D, No. 12, pp. 872-879 (1979).
13) Knuth, D. E.: *The Art of Computer Programming, Vol. 3, Sorting and Searching*, pp. 481-499, Addison-Wesley Pub. Co., Massachusetts (1973).
14) 吉村，武内，津田，首藤：コスト最小法を用いた日本語文の形態素解析，情報処理学会自然言語処理研究会資料，60-1 (1987).
15) 奥村，中村：鉄道信号制御装置に見るフォールト・トレラント・システムの設計，pp. 152-175，日経エレクトロニクス，3-1 (1982).
16) 田中：解析から合成までを融合した英日機械翻訳システム，pp. 275-293，日経エレクトロニクス，8-29 (1983).
17) 古川，後藤，稲垣：LSI の診断に威力を発揮する電子ビーム・プロービング，pp. 172-201，日経エレクトロニクス，3-15 (1982).
18) 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, pp. 140-146 (1982).
19) 首藤，楢原：日本語の文構造のわく組みを与える表現―機能カテゴリーと接続ルール―，福岡大学総合研究所報，Vol. 63, pp. 1-52 (1983).

(昭和63年1月11日受付)
(平成元年1月17日採録)

吉村 賢治（正会員）

昭和30年生．昭和53年九州大学工学部電子工学科卒業．昭和58年同大学院博士課程修了．工学博士．同年同大学工学部電子工学科助手．昭和59年福岡大学工学部電子工学科講師．昭和61年同助教授，現在に至る．自然言語の機械処理に関する研究に従事．電子情報通信学会，日本認知科学会，人工知能学会各会員．

武内美津乃（正会員）

昭和38年生．昭和60年福岡大学理学部応用物理学科卒業．同年同大学工学部副手．昭和62年助手．自然言語処理に関する研究に従事．

津田　健蔵（正会員）

昭和25年生．昭和47年福岡大学工学部電子工学科卒業．同年日本通信工業(株)入社．昭和49年より福岡大学工学部電子工学科教育技術職員，現在に至る．自然言語処理に関する研究に従事．

首藤　公昭（正会員）

昭和18年生．昭和40年九州大学工学部電子工学科卒業．昭和42年同大学院修士課程修了．昭和45年同大学院博士課程単位取得後退学．工学博士．昭和45年福岡大学工学部講師．昭和49年同助教授，昭和57年同教授，現在に至る．昭和55年より1年間テキサス大学言語学研究センター（LRC）客員研究員．自然言語理解，機械翻訳に関する研究に従事．電子情報通信学会，認知科学会，医療情報学会，ACL 各会員．