文節数最小法を用いたべた書き日本語文の形態素解析†

吉村賢治†† 日高達†† 吉田将††

文節内における単語間の連接規則を記述した文法規則を用いるべた書き日本語文の形態素解析では，日本語文としては不適当な解析を含む多くの解析結果が生じる．これらの解析結果から正しい解析を効率的に得る方法として，ヒューリスティックな情報が利用される．従来，この手法としては最長一致法が用いられているが，根拠が明らかでないうえに解析結果に尤度による優先順位をつけることができないという根本的な欠点がある．本論文では，解析結果の文節数によってその尤度を評価する文節数最小法を提案し，この手法に適した表方式の形態素解析アルゴリズムを与える．アルゴリズムの能率は，最悪の場合に必要とするステップ数，メモリ数ともに入力文字列の長さ n に対して $O(n^2)$ である．また，1,000 文の入力文に対して解析実験を行い，文節数最小法の有効性を確認した．その結果，960 文については文節数が最小となる解析に正解が存在し，残り 40 文も一つ文節数が多い解析に正解が存在した．その他，能率，最初に出力される解析結果の誤り率，尤度による順位付けの能力についても最長一致法と比較実験を行った．最初に出力される解析結果の誤り率は，文節数最小法で 7.0%，最長一致法で 12.4% であり，このことも文節数最小法の有効性を十分示している．

1. まえがき

日本語文は単語単位で分かち書きされる英語文などと異なり，複数文節単位で読点などによって分かち書きされる．したがって，日本語文の形態素解析においては文字の並びである入力文を単語の並びとして認定する処理が必要である．この処理には文節内における単語間の連接規則を記述したもの（文節構造規則と呼ぶ）が一般に用いられる．文節構造規則は明確であり，計算機上での実現も容易である．しかし，文節内の構造しか規定していないため，とくに入力文が仮名またはローマ字表記の場合，日本語文としては不適当なものも含むあいまいな解析結果が生じる．これを避けるため従来の仮名漢字変換システムなどでは，入力文を文節単位で分かち書きするなどの制限を課している[1)~3)]．しかし，形式名詞，補助用言，複合語などに関して文節のとらえ方は明確でないため，入力文をこのように制限することには問題がある．

べた書き日本語文を対象とした形態素解析においては，解析結果の尤度をヒューリスティックな情報に基づいて評価することが必要となる．従来，このための手法としては最長一致法が多く用いられているが，根拠が明らかでないうえに，解析結果にあいまいさがある場合の順序付けが文全体に対する尤度によって行われないという根本的な欠点がある．

日本語文の統語規則は，文を構成する文節間の意味的呼応関係を規定する係り受け構造規則と文節を構成する単語の並びを規定する文節構造規則としてとらえることができる[4)]．したがって，文節構造規則を満足する形態素解析の結果が正しい日本語文であるためには，少なくとも係り受け構造規則を満たす必要がある．係り受け構造規則は二つの文節間の係り受け関係を規定する規則であるから，一つの入力文に対して文節構造規則のみを満たす構造が複数個存在する場合には，文節数が少ない構造のほうが多いものより係り受け構造規則を満たす可能性は大きいはずである．

本論文では，このような考えのもとに形態素解析の結果の尤度を文節数を用いて評価する文節数最小法を提案する．以下，2章で文節構造規則を定義し，3章で文節数最小法を用いた形態素解析アルゴリズムを示す．このアルゴリズムは表方式で，入力文字列の長さ n に対して最悪の場合に必要とするステップ数，メモリ数はともに $O(n^2)$ となる．ただし，文節数が最小となる解析だけを求める場合にはともに $O(n)$ となる．4章では，文節数最小法の有効性を確認するために行った実験とその結果を報告する．

2. 文節構造規則

本論文においては接辞は省略して考える．この場合でも自立語に接辞が連接したものを一つの自立語と考えることにより議論の一般性は失わない．また以下の議論では n 長さの入力記号列 s を固定し，先頭から i 番目 $(1 \leqq i \leqq n)$ の記号を $s(i)$ で表す．

単語 w の綴り W(活用語の場合は終止形の綴り)，

† Morphological Analysis of Non-marked-off Japanese Sentences by the Least BUNSETSU's Number Method by KENJI YOSHIMURA, TOORU HITAKA and SHO YOSHIDA (Faculty of Engineering, Kyushu University).

†† 九州大学工学部電子工学科

品詞H，活用情報Kからなる3項系列(W, H, K)をwの単語構造とよぶ．たとえば，単語"たんご"，"呼ば"の単語構造はおのおの次のようになる．

(単語，名詞，一)，(呼ぶ，5段動詞，未然形)

次に文節構造規則を記述するために幾つかの述語を定義する．

[定義 1] 述語 WS, J, E, C

i 入力記号列sの部分列$s(i+1)s(i+2)\cdots s(j)$に対して，$w=s(i+1)s(i+2)\cdots s(j)$なる単語$w$が存在し，$w$の単語構造が$\alpha$であることを述語$WS(i, j, \alpha)$で表す．

ii 単語構造αが自立語の単語構造であることを述語$J(\alpha)$で表す．

iii 単語構造αの単語が文節末の語になりうることを述語$E(\alpha)$で表す．

iv 単語構造α_1の単語w_1と単語構造α_2の単語w_2の連接w_1w_2が文節内で可能なことを述語$C(\alpha_1, \alpha_2)$で表す．

(定義終)

これらの定義を用いて文節を次のように定義する．

[定義 2] 文節構造規則

n長さの入力記号列sに対して，

$i_0(=0)<i_1<\cdots<i_m(=n)$

なる整数$i_0, i_1, \cdots, i_m$と単語構造$\alpha_1, \alpha_2, \cdots, \alpha_m$とが存在して次のi，ii，iii，ivを満たすときsは文節をなすという．

i $WS(i_{l-1}, i_l, \alpha_l)$ $(l=1, 2, \cdots, m)$

ii $J(\alpha_1)$

iii $C(\alpha_{l-1}, \alpha_l)$ $(l=2, 3, \cdots, m)$

iv $E(\alpha_m)$ (定義終)

この文節構造規則は実際には文節をなすための必要条件にしかすぎない．たとえば，格助詞"に"と副助詞"だけ"の場合，

"彼にだけは　負けたくない"

"彼だけには　負けたくない"

のように"にだけ"と"だけに"の連接が可能であるため，"彼にだけにだけには"のような実際には文節と認められないものも文節と認識されることになる．しかし，本論文では日本語として文法的に正しい入力文を解析する立場から，このような場合について考慮しない．

次に，アルゴリズムの記述を簡明にするため上に与えた述語の拡張および定義を行う．

[定義 3] 述語 $\mathfrak{C}$

単語構造α, βに対して$\mathfrak{C}(\alpha, \beta)$であることは，$C(\alpha, \beta)$であるか，または$E(\alpha)$かつ$J(\beta)$であることと等価である． (定義終)

[定義 4] 述語 LB

4変数述語LBを次のように再帰的に定義する．

i $WS(0, j, \alpha)$かつ$J(\alpha)$であることと$LB(0, j, \alpha, 1)$であることは等価である．

ii $LB(i_1, i_2, \alpha, k)$かつ$WS(i_2, i_3, \beta)$かつ$\mathfrak{C}(\alpha, \beta)$ならば$LB(i_2, i_3, \beta, k+\delta(\beta))$である．ここで$\delta$は次のように定義される関数である．

$$\delta(\alpha)=\begin{cases}0\cdots\neg J(\alpha)\\1\cdots J(\alpha)\end{cases}$$

iii 述語LBはi，iiで定義されるものだけである． (定義終)

整数i, kと単語構造αが存在して$LB(i, j, \alpha, k)$のとき，入力記号列sの部分列$b=s(1)s(2)\cdots s(j)$は次数kの左文節列をなすといい，さらに$E(\alpha)$であるような単語構造αが存在するときbは次数kの文節列をなすという．文節列の次数は文節列の文節数に相当する．bが文節列をなすとき，定義4より

$i_0(=0)<i_1<\cdots<i_m(=j)$

なる整数$i_0, i_1, \cdots, i_m$と単語構造$\alpha_1, \alpha_2, \cdots, \alpha_m$とが存在し，

$$\begin{cases}WS(i_{l-1}, i_l, \alpha_l) & (l=i, 2, \cdots, m)\\ \mathfrak{C}(\alpha_{l-1}, \alpha_l) & (l=2, 3, \cdots, m)\\ J(\alpha_1)\\ E(\alpha_m)\end{cases}$$

が成り立つ，このとき3項系列の列，

$(i_0, i_1, \alpha_1)(i_1, i_2, \alpha_2)\cdots(i_{m-1}, i_m, \alpha_m)$

をbの文節構造とよぶ．

[定義 5] 集合 Γ

集合$\Gamma(i)$ $(0\leqq i\leqq n-1)$を次のように定義する．

$\Gamma(i)=\{(i, j, \alpha)\mid i<j\leqq n,\ WS(i, j, \alpha)\}$

(定義終)

集合$\Gamma(i)$を求めることは，単語辞書検索ルーチンに入力記号列sの部分列$s(i+1)s(i+2)\cdots s(n)$を与えて，その最左部分列(prefix)としての単語の単語構造をすべて求めることに相当する．

3. 形態素解析アルゴリズム

形態素解析アルゴリズムは3.1節で述べるパーズ・リスト作成アルゴリズムと3.2節で述べる文節構造抽出アルゴリズムからなる．以下の議論では述語J, C, Eに対応するルーチンは準備されているものとす

る．述語 J, C は単語構造を評価するように定義したが，実際のルーチンでは，述語Jの場合は品詞のみを評価し，述語Cの場合は前の単語の品詞，活用情報と後の単語の品詞の3組を評価すればよい．述語Cは接続テーブルとよぶ2次元マトリクスとして実現される．これらの述語に対応するルーチンの一回の実行に要するステップ数，メモリ数はともにある定数となる．また，集合 $\Gamma(i)$ を求めるルーチン，つまり単語辞書検索ルーチンも準備されているものとする．本論文では単語辞書のデータ構造に関する議論は省略するが，一定のステップ数とメモリ数で $\Gamma(i)$ を求めるルーチンが準備されているものとする．

3.1 パーズ・リストの作成

$0 \leqq i < n$, $0 < k \leqq n$ なる整数 i, k と単語構造 α からなる3項系列 (i, α, k) を項目(item)とよぶ．部分リスト(partial list) $I_j (0 < j \leqq n)$ は項目の有限集合であり，部分リストの全体をパーズ・リスト(parse list)とよぶ．パーズ・リスト作成アルゴリズムを実行した結果，部分リスト I_j に項目 (i, α, k) が属していることは，$LB(i, j, \alpha, k)$ であることを意味している．

[パーズ・リスト作成アルゴリズム]

入力：n長さの記号列 s

出力：パーズ・リスト $I_1, I_2, \cdots, I_n$

初期設定：$I_i = \phi \quad (i = 1, 2, \cdots, n)$

ステップ1

集合 $\Gamma(0)$ を求める．

$\Gamma(0)$ の各要素 $(0, j, \alpha)$ について，$J(\alpha)$ ならば項目 $(0, \alpha, 1)$ を作り部分リスト I_j に加える．

i を1にしてステップ2へ．

ステップ2

$i = n$ ならばアルゴリズムは終了する．

$i \neq n$ のとき，

$I_i = \phi$ ならば，i を $i+1$ にしてステップ2へ．

$I_i \neq \phi$ ならば，

集合 $\Gamma(i)$ を求める．

$\Gamma(i)$ の各要素 (i, j, α) について，

$$\exists i' \exists k \exists \beta . [\mathfrak{C}(\beta, \alpha) \wedge (i', \beta, k) \in I_i] \qquad (1)$$

ならば，式(1)を満たすすべての k の値について項目 $(i, \alpha, k + \delta(\alpha))$ を作り部分リスト I_j に加える．

i を $i+1$ にしてステップ2へ． □

このパーズ・リスト作成アルゴリズムについては以下の定理が成り立つ．

[定理 1] パーズ・リスト作成アルゴリズムを実行した結果，項目 (i, α, k) が部分リスト I_j に属していることは，$LB(i, j, \alpha, k)$ であることと等価である．

証明は i に関する帰納法で容易に行えるので省略する．定理1より，パーズ・リスト作成アルゴリズムを実行した結果，

$$\begin{cases} (i, \alpha, k) \in I_n \\ E(\alpha) \end{cases} \qquad (2)$$

ならば，入力記号列 s は次数 k の文節列をなし，パーズ・リストを参照することにより，その文節構造を抽出できる．パーズ・リスト作成アルゴリズムの能率は次の定理2によって与えられる．

[定理 2] パーズ・リスト作成アルゴリズムの実行に要するステップ数，メモリ数は入力記号列の長さ n に対して，ともに最悪の場合 $O(n^2)$ である．

(証明) メモリ数は項目の総数に比例する．単語の長さと一つの文字列がもつ単語構造の個数は有限であるから，一つの部分リストに含まれる項目の個数は，項目の第3項目が取りうる値の個数に比例し $O(n)$ 個である．部分リストの個数はたかだか n 個であるから項目の総数は $O(n^2)$ 個である．

パーズ・リスト作成アルゴリズムにおいてステップ1は1回，ステップ2は $n-1$ 回実行される．3章の冒頭で述べた仮定からステップ1の実行に要するステップ数は，集合 $\Gamma(0)$ の濃度に比例する．前半の証明と同様に $\Gamma(0)$ の濃度は $O(1)$ である．また，任意の整数 $i (1 \leqq i < n)$ に対してステップ2の実行に要するステップ数は $\Gamma(i)$ の濃度と部分リスト I_i に含まれる項目数の積に比例する．したがって，ステップ2を1回実行するのに要するステップ数は $O(n)$ であるから，パーズ・リスト作成アルゴリズムの実行に要するステップ数は $O(n^2)$ である． (証明終)

ここで，ある関数 $f(n)$ が $O(g(n))$ であるとは，ある定数 c が存在して，$f(n) \leqq cg(n)$ であることを意味している．

3.2 文節構造の抽出

本節ではパーズ・リスト作成アルゴリズムで作成したパーズ・リストを参照して，入力記号列の文節構造を抽出するアルゴリズムについて述べる．入力記号列の文節構造は式(2)を満たす整数 i, k と単語構造 α が存在するときにのみ抽出できる．以下に示すアルゴリズムは，このような項目が存在するとき，指定された次数の文節構造を一つ抽出するアルゴリズムである．文節数最小法に従い次数の小さい文節構造から優先して出力するためには，部分リスト I_n を検索して式(2)を満たす整数 k の値が小さいものから指定して

文節構造抽出アルゴリズムを実行すればよい．部分リスト I_n の検索に要するステップ数，メモリ数はそれぞれ最悪の場合で $O(n), O(1)$ である．

文節構造抽出アルゴリズムでは，計算状況Rとして入力記号列の切れ目を示す整数 i，単語構造 α，次数を示す整数 k からなる3項系列 (i, α, k) を考える．アルゴリズムの実行中に $R=(i, \alpha, k)$ であることは，入力記号列 s の文節構造において，s の部分列 $s(i+1)s(i+2)\cdots s(n)$ に対応する部分を抽出し，最後に出力した3項系列は，ある整数 $j(i<j\leqq n)$ に対して (i, j, α) であり，s の部分列 $s(1)s(2)\cdots s(i)$ は次数 k の左文節列をなすことを意味している．

[文節構造抽出アルゴリズム]

入力：パーズ・リスト $I_1, I_2, \cdots, I_n$，次数 m

出力：入力記号列 s の次数 m の文節構造

ステップ1

$E(\alpha)$ を満たす項目 (i, α, m) が部分リスト I_n に存在するならば，3項系列 (i, n, α) を出力し，Rを $(i, \alpha, m-\delta(\alpha))$ にしてステップ2へ．このような項目が存在しないならば，エラー・メッセージを出力してアルゴリズムを終了する．

ステップ2

$R=(j, \alpha, k)$ とする．

$j=0$ ならばアルゴリズムは終了する．

$j\neq 0$ ならば，$0\leqq i<j$ なる整数 i と $\mathfrak{C}(\beta, \alpha)$ なる単語構造 β と整数 k から構成される項目 (i, β, k) を部分リスト I_j から検索し，3項系列 (i, j, β) を出力したのち，R を $(i, \beta, k-\delta(\beta))$ としてステップ2へ．

□

アルゴリズムの正当性は自明である．アルゴリズムのステップ1および2において，条件を満足する項目が複数個存在する場合にはプッシュ・ダウン・スタック(push down stack)を用いることにより可能な文節構造をすべて出力できる．

定理2の証明と同様にして，文節構造抽出アルゴリズムのステップ2で部分リストから条件を満たす項目を検索するのに要するステップ数は $O(n)$ である．したがって，次の定理が成立する．証明は省略する．

[定理 3] 文節構造抽出アルゴリズムの実行に要するステップ数，メモリ数はそれぞれ入力記号列の長さ n に対して最悪の場合 $O(n^2), O(1)$ である．

定理2，定理3より入力記号列から一つの文節構造を求めるのに要するステップ数，メモリ数は入力記号列の長さ n に対して，ともに最悪の場合 $O(n^2)$ となる．

定義2および定義4から明らかなように，本論文で定義した文節列は非決定性有限オートマトンで受理される．非決定性有限オートマトンは，それと等価な決定性有限オートマトンに変換することが理論的には可能であるが[5]，文節列の場合には状態数が多くなるため実際問題としては困難である．本論文で定義した文節構造の解析はステップ数，メモリ数ともに $O(n)$ の能率で可能であるが[6]，本論文で示したアルゴリズムでは文節数最小法に必要な次数の情報を付加したため $O(n^2)$ になっている．

3.3 最長一致法との比較

本論文で定義した文節構造規則に基づいた解析を最長一致法を用いて行う場合，解析のアルゴリズムはバックトラック方式のアルゴリズムとなる．その能率は入力記号列の長さ n と定数 c に対して，最悪の場合に必要とするステップ数，メモリ数はそれぞれ $O(c^n)$, $O(n)$ である[7]．

従来，最長一致法は多くのシステムに用いられているが，二つの問題点が挙げられる．第一の問題点は，単語または文節を認定する段階で複数の候補が存在する場合に，入力文字列における長さが長いものに高い優先順位を与えることの根拠が明らかでないことである．第二の問題点は，単語または文節の長さという局所的な評価をしているために，文全体に対する尤度の評価ができないということである．また，バックトラック方式の解析アルゴリズムを用いるために，複数個の解析結果が存在する場合に，それらを尤度に従って順序付けすることができず，一度解析を誤ると次善の解析を得ることができないという根本的な欠陥をもっている．

4. 実　　験

本章では，文節数最小法の有効性を確認するために行った実験とその結果について述べる．実験は同一の入力文に対して最長一致法についても行い，能率，解析結果の品質について両者の結果を比較，検討する．以下の実験システムは，九州大学大型計算機センターの FACOM-M 200 上に PL/I を用いて実現した．

4.1 文法規則

実験には，文法規則として定義2の文節構造規則を用いた．ここで，述語Cは130×375のビット・マトリクスとして実現されている．

4.2 アルゴリズム

文節数最小法の解析アルゴリズムは，3章で示した表方式のアルゴリズムを用いた．ここで，同一文節数で複数の解析結果が存在する場合には，次の手順で優先順位を決定した．

ⅰ 自立語の次の語は付属語を，付属語の次の語は自立語を優先する．

ⅱ 活用語を優先する．

評価の順序はⅰ，ⅱの順である．ⅰは日本語文の統計的性質[8]に基づく評価であり，ⅱは単語の使用頻度を近似した評価である．

最長一致法の解析アルゴリズムは，プッシュ・ダウン・スタックを用いたバックトラック方式のアルゴリズムを用いた．アルゴリズムの詳細は文献7)に譲る．アルゴリズムにおいて，入力記号列のある位置から始まる単語を認定する段階で複数の候補が存在する場合の優先順位は次の手順で決定した．

ⅰ 自立語よりも付属語を優先する．

ⅱ 入力記号列における単語の長さが長いものを優先する．

ⅲ 活用語を優先する．

ここで評価の順序はⅰ，ⅱ，ⅲの順である．

4.3 単語辞書

実験に用いた単語辞書は自立語辞書と付属語辞書からなる．自立語辞書には約83,000語の自立語が登録されており，付属語辞書には約300語の付属語，形式名詞，補助用言が登録されている[9]．

4.4 入力文

実験には入力文として武者小路実篤の「人生論」から最初の1,000文を用意した．入力文はJIS漢字コードによる平仮名表記のべた書きであり，一文の平均長さは44文字で，20文字程度に読点で分割されている．3章で示したアルゴリズムは適当に文節間にブランクなどを挿入した入力文も処理できるが，実験に用いた入力文は原文中の読点以外のブランクなどによる分割はされていない．

4.5 文節数最小法の有効性

1,000文の入力文について実験した結果，960文は文節数が最小となる解析のなかに正解が存在した．ここで正解とは入力文の原文と一致する解析である．ただし，原文との比較は品詞，活用情報についてのみ行い，綴りの区別はしないものとする．たとえば，入力文"きかい"に対しては"機械"，"機会"などの名詞が対応するが，これらの区別はしていない．文節数が最小となる解析に正解が得られなかった原因となった部分を表1に示す．表において"|"は文節の切れ目を示している．原文"その人"の正解が文節数の最小となる解析中に得られなかった原因は，本来2文節である"その日"が一つの単語として自立語辞書に登録されていたことにある．ヒューリスティック情報として文節数最小法を用いる場合には，このような複合語に対しては文節数に相当する荷重情報が必要である．表1からも明らかなように実験した1,000文に対しては，正解は文節数が最小となる解析か，その次に文節数が少ない解析にすべて含まれている．このことは文節数最小法の有効性を十分示している．

文節数が最小となる解析では，入力文1,000文で平均して，入力文字列10文字当り2通のあいまいな解析が存在する．

表1 文節数が最小となる解析が正解でなかった原因
Table 1 The causes that no correct analysis was found in the analyses that have the least BUNSETSU's number.

原文	解析	件数
その｜人	その日と	13
良く｜して	浴して	5
子が｜ない	漕がない	2
気が｜する	起臥する	2
～の｜ない｜人	～の｜内皮と	2
この｜蚊の	子のかの	1
その｜産む｜能力	圃生｜無能力	1
死｜その｜もの	始祖の｜もの	1
見て｜嫌に	未定やに	1
もう｜あとは	盲唖とは	1
具合｜良く｜育って	具合よ｜養だって	1
目｜医者	名車	1
本屋が｜良い｜本を	本屋がよ｜異本を	1
招き｜易い｜点だ	招きや｜水天だ	1
そう｜たいした	早退した	1
しかし｜金と	鹿しかねと	1
どうか｜すると	同化すると	1
～｜書く｜気は	～か｜茎は	1
どう｜言う｜人を	同意｜雨飛とを	1
～の｜見た｜美が	～のみ｜旅が	1
男が｜何か	男仮名にか	1

(1,000文中)

表1から明らかなように実験した1,000文に対しては，正解は文節数が最小となる解析か，その次に文節数が少ない解析にすべて含まれている．このことは文節数最小法の有効性を十分示している．

文節数が最小となる解析では，入力文1,000文で平均して，入力文字列10文字当り2通のあいまいな解析が存在する．

4.6 能率の比較

文節数最小法と最長一致法において，解析に必要としたメモリ数の比較を図1に示す．横軸は入力文の長さ，縦軸はメモリ数である．文節数最小法(LBN)は最長一致法(LM)の2倍のメモリ数を必要としているが，入力文の長さが80文字のとき必要とするメモリ数は約30 kbyteであり，現在の計算機の規模を考えるとそれほど大きい値ではない．

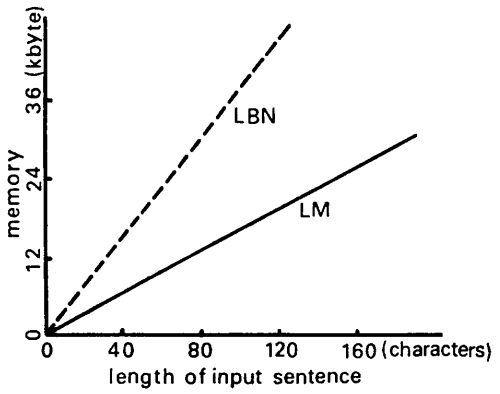

図 1 メモリ数の比較
LM：最長一致法，LBN：文節数最小法
Fig. 1 Comparison of required memory size. LM : The longest match method, LBN : The least BUNSETSU's number method.

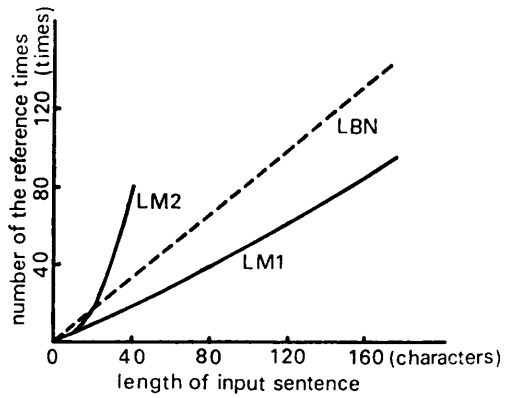

図 2 自立語辞書の検索回数．LM 1：最長一致法(第一番目の解析を求める場合)，LM 2：最長一致法(正解を求める場合)，LBN：文節数最小法
Fig. 2 The number of the reference times. LM 1 : The longest match method. (In the case of getting only the first analysis) LM 2: The longest match method. (In the case of getting the correct analysis) LBN : The least BUNSETSU's number method.

現在の計算機技術においては，本論文で述べた形態素解析に必要とする処理時間の大半は二次記憶上にある自立語辞書の検索に費やされる．そこで，実験ではステップ数の比較に代えて自立語辞書の検索回数を比較した．その結果を**図 2** に示す．横軸は入力文の長さ，縦軸は自立語辞書検索ルーチンの呼び出し回数である．最長一致法において一番目の解析だけを求める場合(LM 1)は文節数最小法 (LBN)より優れているが，正解を求めるまで (LM 2) に行う自立語辞書の検索回数は，入力文の長さに対して指数関数的に増加している．

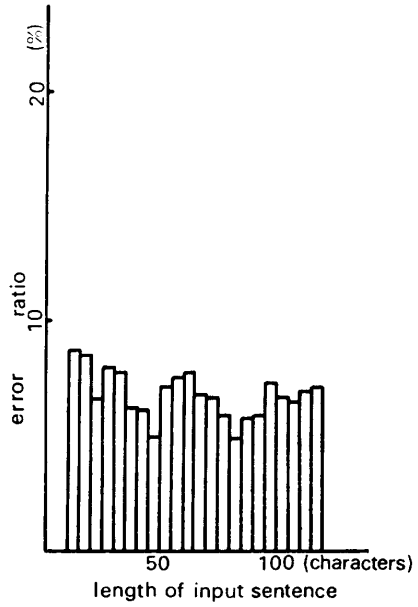

図 3 最初に出力された解析の誤り率 文節数最小法
Fig. 3 Error ratio of the first analysis. The least BUNSETSU's number method.

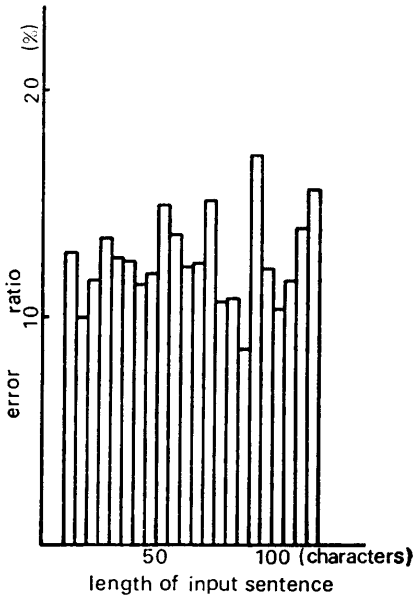

図 4 最初に出力された解析の誤り率 最長一致法
Fig. 4 Error ratio of the first analysis. The longest match method.

4.7 解析結果の品質の比較

解析結果の品質の比較として，ここでは一番目に出力される解析結果の誤り部分の入力文字列における長さを比較する．**図 3** に文節数最小法，**図 4** に最長一致法の結果を示す．それぞれ横軸は入力文の長さ，縦軸

表 2 正解に与えられる順位
Table 2 The given order of the correct analysis.

	文節数最小法(文)	最長一致法(文)
1位	124	73
2位	28	3
3～4位	20	8
5～8位	23	4

(300 文中)

は誤り部分の入力文字列における長さの入力文字列の長さに対する割合である．解析結果に対する正誤の判定は 4.5 節と同様な基準で行った．誤り率の平均は，文節数最小法で 7.0%，最長一致法で 12.4% である．

4.8 尤度の評価

3.3 節において，最長一致法は文全体に対する尤度による優先順位付けが正しくできないことを述べた．ここでは，尤度による優先順位付けの能力を比較するために，文節数最小法，最長一致法それぞれの手法によって正解に対して付けられた順位を入力文 300 文について比較する．その結果を**表 2** に示す．表 2 からも明らかなように，最長一致法では一度解析を誤ると容易に正解に到達しない．なお，4.5 節で述べたように文節数最小法での優先順位が 8 位以内の解析は，ほとんど文節数が最小となる解析に含まれている．

5. むすび

4 章で示した実験結果より，必要とするメモリ数，第一番目の解析を求めるのに要する処理時間の点で最長一致法が優れている．しかし，正解を求めるまでに要する処理時間や一番目に出力される解析結果の品質，尤度の評価能力などについては文節数最小法が優れている．これまで日本語文の形態素解析において，解析結果の尤度を評価する手法のほとんどは最長一致法であったが，文節数最小法も同等またはそれ以上の能力をもっている．日本語文の形態素解析において残されている重要な課題である未登録語，複合語，接頭・接尾語の処理の問題を考える際に一技法としてその利用法を検討する必要がある．

また，3.3 節において最長一致法は明確な根拠をもっていないことを述べたが，局所的な単語または文節の長さをできるだけ長くとることによって文全体の文節数を最小に近づけようとしていると考えることもできる．しかし，本論文で提案した文節数最小法と最長一致法は原理的に異なるもので，前者では解析結果の全体に尤度による順位付けが行われるのに対し，後者ではそれが行えない．

謝辞 本実験に使用した自立語辞書の作成に初期のころから従事し，以来十数年間これに専念してこられた九州芸工大稲永紘之講師，小西彬充助手をはじめ，実験システムの実現のために多くの方々の協力をいただいた．深く感謝の意を表する．なお，本研究は昭和 52～54 年度文部省科学研究費特定研究「言語」，昭和 54～55 年度文部省科学研究費試験研究(1)「かな漢字変換を中心とした日本語入力システムの開発」および昭和 56 年度試験研究(2)「効率的日本語単語辞書の作成」および 56～57 年度 21 世紀文化学術財団学術奨励金によった．

参考文献

1) 栗原，黒崎：仮名文の漢字混り文への変換について，九大工学集報，Vol. 39, No. 4, pp. 659-664 (1967).
2) 相沢，江原：計算機によるカナ漢字変換，NHK 技研，Vol. 25, No. 5, pp. 23-60 (1973).
3) 森，河田，天野，武田：計算機への日本語情報入力，信学技報，EC78-23, pp. 33-41 (1978).
4) Hitaka, T. and Yoshida, S.: A Syntax Parser Based on the Case Dependency Grammar and Its Efficiency, *Proc. COLING 80*, pp. 295-302 (1980).
5) Hopcroft, J. E. and Ullman, J. D. (野崎，木村訳)：言語理論とオートマトン，p. 30，サイエンス社，東京(1976).
6) 吉村，日高，吉田：日本語の構文分析—文節の統語規則と表方式を用いた文節構造分析アルゴリズム—，九大工学集報，Vol. 54, No. 1, pp. 35-39 (1981).
7) 吉村，日高，吉田：最長一致法と文節数最小法について，情報処理学会人工知能と対話技法研究会資料，24-1 (1982).
8) 牧野，木澤：べた書き文のカナ漢字変換—文節形による分かち書き—，信学技報，PRL 77-27, pp. 65-72 (1977).
9) 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).

(昭和 57 年 4 月 16 日受付)
(昭和 57 年 6 月 15 日採録)