自然言語処理 27－3
（1981. 9. 24）

自由入力形式のカナ漢字変換

内田裕士，杉山健司（富士通研究所）

1．はじめに

日本語を計算機に入力するための有効な手段として，カナあるいはローマ字で日本語文を入力し，計算機によって自動的に漢字カナ混じり文に変換するカナ漢字変換が古くから研究されてきた．この方式は入力機器として特別の装置が必要でなく，また誰でも手軽に扱えるという点から，素人向きの日本語入力手段として普及している．

カナ漢字変換の初期の研究においては，入力する文にあらかじめ分ち書きを施したり[1),2),3)]，制御記号を挿入する方式が提案されてきた[4),5)]．しかしながら，入力単位に制限を設けることは，オペレータの負担となって入力速度の低下を招いたり，何かを考えながら入力する時に思考の妨げとなったり，さらには，分かち書きのしかたの個人差のためにカナ漢字変換に悪影響を及ぼすなどの欠点があった．

これらの欠点を取り除くべく，ベタ書き文を許容するようなカナ漢字変換システムや[6),7)]，文節の概念をより柔軟にすることによって，入力単位の制限をゆるめようとする研究が最近でてきている[8),9)]．本稿で述べるカナ漢字変換方式は，これらの研究の流れに沿ったもので，ベタ書きを含む入力単位に制限を持たないものである．

本稿では，入力単位に制限を持たないカナ漢字変換方式を提案する．第2章においては，入力に制限がない自由入力形式について説明し，第3章においては，最尤評価法と木探索によるカナ漢字変換アルゴリズムについて，第4章では，カナ漢字変換のための文法について述べる．第5章では，実験結果について述べる．

2．自由入力形式

自由入力形式とは入力する文の単位に全くといっていいほど制限のない入力形式のことをいう．入力に対する唯一の制限は，1つの単語を構成するカナ文字列は分かち書きしないという程度である．すなわち，「社会の高度化に従い．．．」という日本語文を入力する場合，以下のどのような入力形式をとってもよい．

1）シャカイ ノ コウド カ ニ シタガ イ （単語分かち書き）

2）シャカイ ノ コウドカ ニ シタガ イ （自立語，付属語分かち書き）

3）シャカイノ コウドカニ シタガイ （文節分かち書き）

4）シャカイノコウドカニシタガイ （ベタ書き）

5）シャカイ ノ コウドカニ シタガイ （単語分かち書きと，文節分かち書きの混合）

6）シャカイノコウドカニ シタガ イ （ベタ書きと，単語分かち書きの混合）

上の例では自立語から入力が始まっているが，自由入力形式の場合，入力は付属語から始まってもよい．このような自由さは他のカナ漢字変換システムには全く見られない．旧来の多くのシステムでは[1)~5)]，入力に制御コードを挿入したり，分かち書きをするなど，固定的な制限を設けているし，最近のシステムにおいても[6)~8)]，必ず自立語から始まらなければならないという制限がある．この制限は，入力を行う者にとってかなり自然なものであるという暗黙の仮定からきていると思われる．しかしながら，文法知識に乏しいオペレータによる入力を考えた場合，この仮定は少し無理があるように思われる．例えば，「～しなければならない～」という文において，「いう」は自立語であるのかどうかを決定するのは一般に困難である．多くの文法書では，「いう」は補助用言として付属語に分類されているが，これらは一般常識ではないからである．このような，自立語，付属語の区別のあいまい性の他に，実際のテキストの作成に際して，自立語のあとに付属語をつけて入力することは必ずしも一般的で

はないと考えられる．例えば，「制限がある」という日本語文を入力するとき，「セイゲン」を入力してしまってから，「ガアル」という付属部があることに気がつくこともあろうし，何かを考えながら入力するときはなおさら自立語と付属語を別々に入力したくなるであろう．

このように入力操作の使い易さや柔軟性を追及すれば，ここでいう自由入力にたどりつく．この方法では単語の区切り目が分かればそこで区切ってよいし，区切らずに続けて入力してもよい．このような自由な入力形式を可能にするために，本システムにおいては，自立語，付属語といった概念はなく，ただ単語という概念があるのみである．

3．木探索と最尤評価法

1）木探索

自由入力形式のかな漢字変換を実現するために，木探索手法を応用して入力列から単語を抽出している．例として「キノウデシゴトガ」という仮名文字列からの単語抽出を考える．この場合，図1のような木探索によって単語抽出を行う．この解析木のノードは候補単語であり，アークは文法チェックを表わす．また図の右側にあるノード程レベルあるいは深さが深いといい，成功と指示されたノードにいたる経路を解と呼ぶ．

まず，入力列「キノウデ．．．」の左端から始まる文字列と，単語辞書とのマッチングを行い，候補単語「昨日」，「木」，「黄」などを見付ける．単語辞書には単語の読み，書きのほか，文法に関する情報や，出現頻度に関する情報が格納されている．

ここで見つかった候補単語の中から一番適当と思われるものを決める．この際，単語の長さや頻度が参照される．従来，文節からの単語抽出に使用されてきた最長一致法は語長が長いほど妥当なものとし，同じ語長の中では頻度が高いものを妥当なものとする．いま，この手法に従うとすると，「昨日」が一番妥当性がある．そこで一旦「昨日」が正しい単語であると仮定し，初期入力列から「昨日」の読みである「キノウ」を除いた文字列「デシゴト．．．」について解析を進める．

前段階と同様にして，辞書とのマッチングを行い，候補「弟子」，「で」を捜し出す．最長一致法に従って「弟子」を優先し，「昨日」と「弟子」との接続チェックを行う．この場合は接続可能なので，「弟子」を除いた入力列「ゴトガ」についてさらに解析を進める．「弟子」を抽出後，候補として，「事」を見付け出すが，「弟子」との接続チェックの結果，接続不可能であることがわかるので，1つ前に戻る．すなわち，前段階で「弟子」が正しいものであると仮定したのが誤りであったと考え，「弟子」の次に有力な候補「で」以降を調べる．木探索の途中で前の方へ戻る事をバックトラック制御という．「で」が正しいものであるとして，上記と同じような辞書検索と接続チェックを繰り返し，全入力列を分解し終えた時，探索を終了する．いまの例では，「昨日－で－仕事－が」が変換結果として得られる．

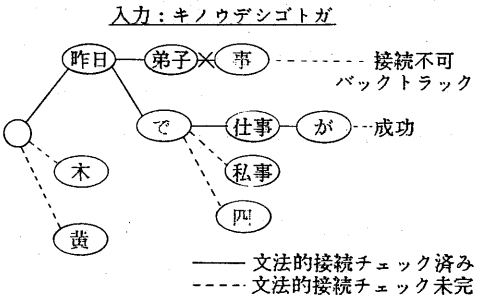

図1　単語抽出木探索

2）最尤評価法

図2の例で，「昨日」，「木」，「黄」などの候補のうち，どれが有力かを決める場合，従来は最長一致法が使われてきた事はすでに述べた．しかしながら，これをそのまま自由入力形式の単語抽出に応用することは，助詞，助動詞のように出現頻度が高い割りには単語長が短いものが，他の単語に比べて常に優先度が落ちる事になり，必ずしも適当な方法ではない．例えば，図2の深さ2の探索において，「で」より「弟子」の方が優先され，ムダな探索（結果としてバックトラックがかかった）を

してしまう．この場合はバックトラックの結果，正解が得られたが，誤った選択を行ったまま，探索を終了してしまう事も多い．例えば図２の例で単語辞書に「毎」などがはいっている場合は，「弟子」と「毎」は接続し，「昨日弟子毎が」が変換結果となる．ここで「で」の方が優先されれば，バックトラックもかからず，より早くゴールに到達することができるし，また正解がえられることにもなる．

本稿で提案している最尤評価法は，候補単語の優先度を決定するのに，語長と頻度を組み合わせ，最長一致法より一歩進んだ方式となっている．最尤評価法では，優先度を決定するのに次のような評価関数を使い，この評価値が大きいほど，優先度が高いと見なす．

$$F(W) = F(|W|, Hw) \quad (1)$$

ただし，$|W|$ ：単語Wの読みの長さ

Hw ：単語Wの頻度

この評価関数は語長が長ければ長いほど，また頻度が高ければ高いほど大きな値になるように決められている．

従来使われてきた最長一致法は評価関数が次の形をした最尤評価法の一種であるといえる．

$$F(W) = W + \frac{Hw}{A} \quad (2)$$

ただし，$|W|$ ：単語Wの読みの長さ

Hw ：単語Wの頻度

A ：$(W \in D \Rightarrow Hw < A)$ を満たす定数（Dは辞書内の単語の集合）

この形の評価関数では語長が短い単語の優先度は常に低く，評価が固定化し，先程述べたムダな探索や，誤った選択が行われ誤った結果を導き出す．そこで，式（1）のような一般化を行い，頻度が非常に高い単語は語長が短くてもある程度優先されるようにすることで，この種の欠点を補うことができる．

最尤評価法による単語抽出を図示すると図２のようになる．候補W_1，W_2，．．．．，W_n に対して式（1）で定義された評価関数を適応し，その評価値C_1，C_2，．．．．，C_n による優先順をつける．図ではW_2 が最優先され，W_2 以降の解析が行われる．

われわれの開発したカナ漢字変換システムでは次のような形の評価関数を使用している．

$$F(W) = Hw \cdot A^{|W|} \quad (3)$$

ただし，$|W|$ ：単語Wの読みの長さ

Hw ：単語Wの頻度

A ：カナ文字の種類の数

ここでＡは一応の目安としてカナ文字の種類の数に決定されている．このように決めればＡがカナ文字列にマッチングするマッチング確率となり，出現頻度と掛け合わせることで評価関数としてうまく働くことが期待される．

この評価関数の意味は，読みの長さが１だけ少ない単語でも，その頻度がマッチング確率の差を超える（頻度がＡ倍）なら，その単語の方が優先されるということを示している．だたこれは一応の目安であって，実際に得られる変換率には，頻度の収集の程度や後で述べる木探索の打切り条件などが影響してくるので，最適値は実験によって確かめる必要がある．この実験結果は第５章において示す．

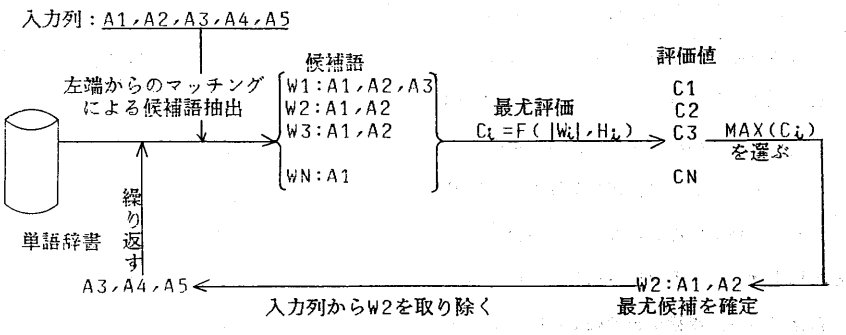

図２　最尤評価法による単語抽出

3）木探索における打切り条件

前述した木探索においては，入力列をすべて解析し終えてから抽出結果をだしている．しかし，一般にはすべての単語が辞書に登録されているとは限らず，前述の方法では入力文字列中に１つでも未登録語があると解析不能になったり，探索木の浅い部分で正しく単語抽出をしていても，その部分までバックトラック制御によって壊れてしまう恐れがある．

また，入力列が長くなれば指数関数的に探索すべきノード数が増し，ある時間内に探索を終了することが不可能となる．そこで木探索において打切り条件を設け，探索途中でも打切り条件を満たした場合は，候補単語の確定を行い，探索木から選択の可能性をある程度消去するようにした．打切り条件は次の２種類がある．

（1）最小深さ，最低読み長条件

（2）字種による打切り条件

最小深さ，最低読み長条件

最小深さ条件とはある決められた深さより浅いか，または，等しい間は可能な限り探索を続け，候補単語の確定を行わないことである．この深さが，あまりに浅いとチェックが甘すぎ，誤った確定を行う可能性が増し，あまり深いと，先程述べたように，未登録語に弱く，変換速度も落ちてしまう．

最低文字列長条件は，次のような現象を防ぐために導入された．すなわち，日本語には読みが１字の名詞（例えば我，位，鵜．．．）が多く，また一般に名詞は名詞とくっついて連語をつくり得るので，ほとんどどんな入力列でも，１語名詞の連鎖として解析することができる．しかし，このような連鎖で文や句が構成されることは実際にはない．先に述べた最小深さだけでは１語名詞の連鎖で最小深さを超える単語抽出が可能である，そのために変換誤りが生じやすい．そこで打切り条件として探索の深さだけでなく，それまで探索された単語（祖先ノード）の読み長の和を考え，ある決められた長さ以下の間は可能な限り探索を続け，候補単語の確定は行わないようにした．この長さが短すぎても長すぎても最小深さ条件と同様の不都合が起こる．従って，この最小深さと最低文字列長は注意深く決めなければならない．われわれの開発したシステムにおいては最小深さ３，最低文字列長７としている．

図３は，最小深さ３，最低文字列長７という打切り条件による単語の確定を示す（図において，「弟子」より「で」が優先されるのは最尤評価法による．）「が」まで探索が進んだ時の深さおよび自分自身を含んだ先祖ノードの文字列長の和はそれぞれ４，８であるので，打切り条件が成立し，単語の確定が行われる．この条件による単語の確定はゴールノードに到達したときの単語の確定とは異なり，確定される単語は「が」の祖先である深さ１のノード「昨日」だけである．

「昨日」を確定後，探索木は図３に示すように，確定された単語と同じ深さにあった単語は消され，確定された単語が新たにルートノードとなる．そして「が」以降の探索を進めて行き，「終」，「尾」を見付け出し，そのうち優先度の高い「終」を取り出し，文法チェックを行う．この「終」の深さおよび解析済みの文字列長はそれぞれ４，７で最小深さ条件は満たすが，最低文字列長条件を満たさないので，この段階では単語の確定は行われず，さらに深く探索が進む．入力列がすべて解析しつ

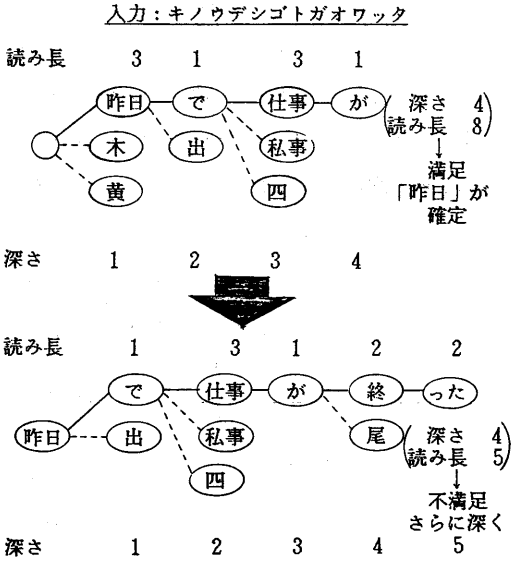

図３　深さ，読み長による打切り条件

くされるまで，同様のことが繰り返される．入力列の最後まで解析し終えたときは，図1と同じようにゴールノードをとおるパス上の単語すべてを確定する．

字種による打切り条件

深さ，文字列長条件以外に字種による打切り条件がある．このような字種は，文法的に違いのあるものに関して入力文字列に対してただ1つの候補しかないことが条件となる．句読点や英数字などはこれらの条件を満たしている．図4のように「キノウデ」の後ろに読点があるとする．読点はそれ以外の解釈のしようがないので，それまでの解釈がすべて正確であったと考えそのパス上にあるすべての単語を確定する．これらの特別な字種はゴールノードと同じような働きをする．

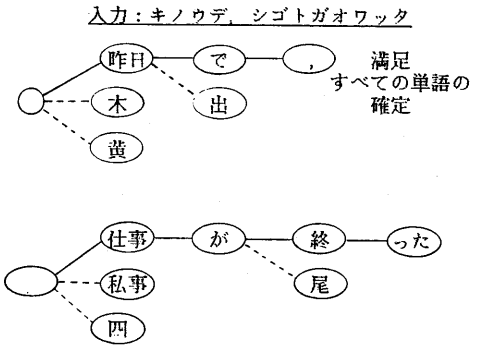

図4　字種による打切り条件

4）探索不能時の処理

先に述べたように探索はゴールノードに到達する（入力列のすべてを解析し終える）か，打切り条件が満たされるまで続くが，未登録語があったり，入力ミスがあったりすると，どうしても条件が満たされず，バックトラック制御によりルートノードまで制御が戻る場合がある．このときは，探索が失敗したということであり，すべての可能性をつくしたが解析できなかったことを意味する．そこで最初の1文字を入力列から落とし，以後の入力列に対して同様の探索を試みる．最初の1文字はかなをひらがなに変換して出力する．探索不能時の処理を図5に示す．

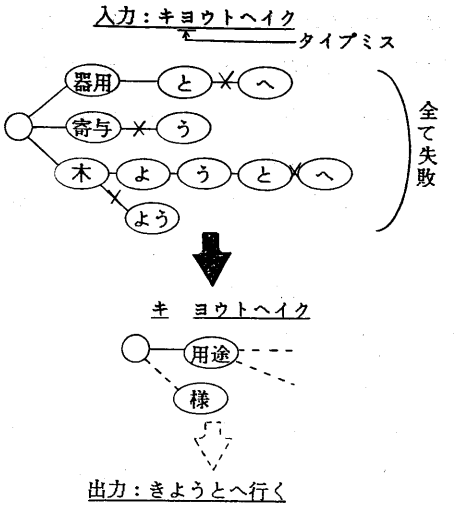

図5　探索不能時の処理

5）枝刈り

図6のような例を考える．入力列「ハシル」に対して，「橋」，「端」など多くの候補がある．これらの候補に図の上から下へ並んでいる順番に優先順位がつけられているとする．まず「橋」が有力候補としてとられ，「る」との接続チェックが行われる．これは接続不可能となりバックトラックがかかり，次の候補「端」がとられる．ここで以前までは単純に「端」以降の入力列「ル」について解析を行ってきた．しかし，その必要があるだろうか．「橋」が候補からはずされた（バックトラックがかかった）理由は「橋」と「る」が文法的に接続しえないことにある．「端」は「橋」と読みが同じで，かつ品詞も同じである．従って，「端」以降の解析をしなくても

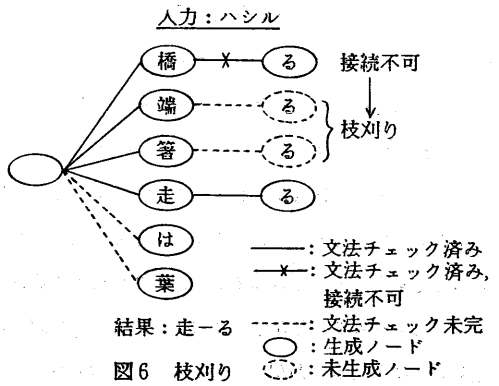

図6　枝刈り

「橋」がはずされたのと同じ理由で「端」も候補として不適当であることがわかる．さらに「箸」も同じ理由で不適当である．これら不適当な候補を探索からはずすことによってムダな探索をせずにすむ．

このように候補の探索からムダとわかった候補を探索対象からはずすことを（後向き）枝刈りと呼ぶ．この枝刈りを行える条件は同じ深さで同じ読み，同じ文法属性を持っていることである．

4．二方向文法

われわれはカナ漢字変換のために各々の単語が持つべき文法特性を考えるに当たって，必ずしも既存の日本語の文法分類に固執することなく，カナ漢字変換に適するよう，単語間の接続性に着目した文法分類を行った．この際，特に問題としたことは，

1）単語間の接続の可否を表わす接続関係表（行列）をコンパクト化すること，

2）カナ漢字変換の変換率や変換効率を上げるために，熟語や成句を入れやすくすること，

であった．

この問題を突き詰めていくと必然的に，単語の接続性を右と左に分離して考える，二方向文法という考えに到った．これは単語の持つ文法特性を，その接続性という観点から考えれば，単語の右に接続可能な単語のグループと，左に接続可能な単語のグループは当然違ってくるので，単語は左右に別の顔（文法特性）を持っているであろうし，接続文法特性を左右に分離して考えれば，まとめて考えるよりも，文法分類は，当然減少するので1）に上げた問題も解決されるからである．

また，熟語や成句を辞書に入れるためには，この単語に文法特性を与えなければならないが，一般にこの文法特性を決めることはなかなか困難である．例えば，「日本語」や「によって」という成句に対して文法特性を与えることを考える．「日本語」に対しては通常の名詞としてとらえられる．しかし，「によって」という成句は，格助詞「に」とも，動詞「よっ」とも，接続助詞「て」とも異なる．ここで，「によって」の接続性に注目してみれば，図7に示すように，「によって」という成句の左の接続性は格助詞「に」と同じであり，右の接続性は接続助詞「て」と同じであることにきがつく．従って，二方向文法を採用すれば，熟語や成句の文法特性は機械的に決めることが出来，しかもその際に，接続関係表が大きくなる心配はないので，2）に上げた問題も解決される．

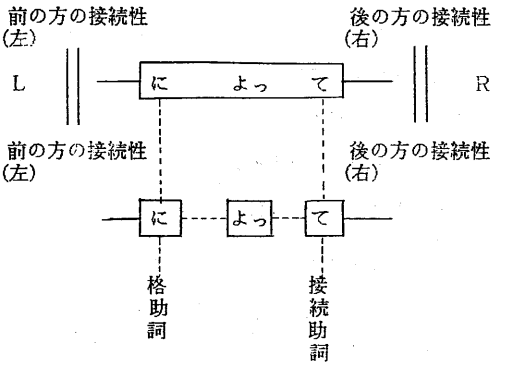

図7　成句の接続性

二方向の文法特性と単語間接続関係表（行列）の関係は，右文法特性Rは接続行列の行番号を示す値となり，左文法特性Lは列番号を示す値となる．

ここで，さらに，この接続行列をコンパクトにすることを考える．「1」や「2」といった数を表わす数詞と「個」，「冊」といった助数詞との関係を考える．助数詞の左文法特性を考えると，ただ1種の品詞の単語，すなわち数詞だけが前にくるだけである．これを接続行列上でみると図8のように助数詞の左文法特性の列には，1（接続可能を示す）がただ1つあるのみである．このような特性を持った左文法特性のものとしては，接続詞や動詞，形容詞の活用語尾など比較的多くのものが考えられる．これらの接続関係を図8のように表現しておくことは，0が多くいかにもムダである．そこで補助右文法特性というもう1つ別の接続情報を辞書項目に付加し，仮想的な行列表現を行うことにした．

図9にあるように実装された行列の右に仮想的な行列の領域があり，その仮想空間上の1点を示すために補助右文法特性を使う．この補助文法特性は左右の文法特性

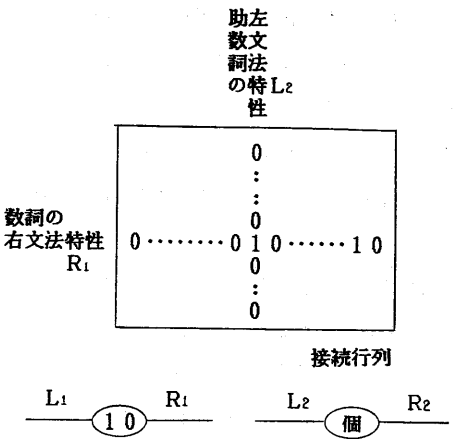

図８　特異な接続関係

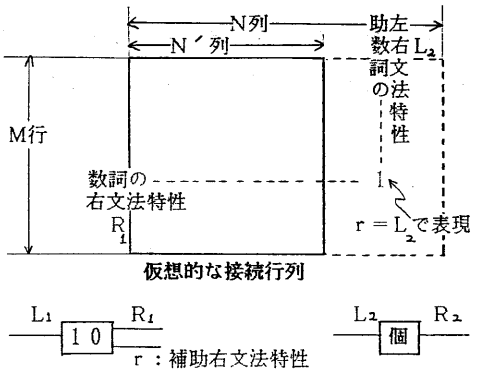

図９　補助情報による仮想行列表現

とは異なり，接続行列の行や列を指定するものではなく，ある行列上の位置が１であることを示す．こうすることによって本来Ｍ＊Ｎ行列であった接続行列をＭ＊Ｎ′（Ｎ′＜Ｎ）とすることができる．

さらに，補助文法特性は仮想行列の１点を示せるだけでなく，実装行列上の１点をも示すことができるので，行の接続情報が実装行列内で１点しか違わないものをまとめることができ，接続情報のコンパクト化が図れる．

一般に，カナ漢字変換の能力を高めるためにはより長い単位で辞書に登録するほうがよい．しかし，長い単位で登録すれば，それだけ多くの項目を辞書に入れなければならなくなる．従って，二方向文法により熟語や成句などの長い単位で辞書作成が可能になったが，どういう種類の熟語や成句を辞書に登録するかを注意深く検討しなければならない．そこでカナ漢字変換の能力を高め，しかもむやみに辞書項目数が増えないようにほぼ以下の方針に従って考えた．

１）助詞や助動詞，補助用言などの付属語は，文字数も短く誤変換のもとになりやすい．しかもよくいわれるようにこれらの単語は数が少なく，組み合わせもかなり制限されているので，なるべくこれらは成句として長い単位で辞書に登録する．

２）名詞や副詞などの自立語は，数も多く，また文字数も比較的長いのでこれらを含んだ成句を登録することは避ける．

このような付属語・自立語にたいする基本的な登録方針をもとに，二方向文法の特徴を生かし，なるべくコンパクトな接続行列を実現するため，従来の国文法の品詞分けを整理・統合して最終的なカナ漢字変換のための文法体系を作った．

５．評価

本稿で提案したカナ漢字変換方式の有効性を調べるため，誤変換率の測定を行った．ここでいう誤変換率は，誤変換された文字数を正解の文字数で割ったものである．

最尤評価をうまく機能させるためには，第３章で述べたように式（３）の定数Ａをうまく決める必要がある．実験では評価速度を考慮して，Ａを２のべき乗とし，

$$A = 2^{\alpha} \qquad (4)$$

とし，この式のαをいろいろ変化させて実験を行った．また，木探索をせず最尤評価のみでカナ漢字変換を行った場合や，常に一番長い単語を優先し，同じ長さの単語同士の選択に頻度情報を用いる最長一致法による変換についても比較のために実験を行った．その結果を図１０に示す．

この実験で用いたデータは７４５０文字の新聞記事であり，分野は政治，経済，社会を含んでいる．入力は全くのベタ書きであり，辞書には約１０万の単語が登録されている．

この実験により，木探索を用いない最尤評価法だけでも最長一致法よりも能力が高いことがわかる．また，最尤評価の定数としては，$\alpha = 5$が最適であることがわか

った．

6．おわりに

本稿で提案したかな漢字変換方式は，現在，富士通のＪＥＦのアプリケーションであるＦＤＭＳおよび右筆に組み込まれ，ユーザに提供されている．

参考文献

1）相沢，江原：計算機によるカナ漢字変換，ＮＨＫ技術研究，Ｖｏｌ．２５，Ｎｏ．５，ｐｐ．２６１－２９８（１９７３）

2）木村，遠藤，小橋：日本語文入力用カナ漢字変換システムの試作，情報処理，Ｖｏｌ．１７，Ｎｏ．１１，ｐｐ．１００９－１０１６,（１９７６）

3）牧野，勝部，木沢：カナ漢字変換の一方法，情報処理，Ｖｏｌ．１８，Ｎｏ．７，ｐｐ．６５６－６６３（１９７７）

4）松下，山崎，佐藤：漢字かな混じり文変換システム，情報処理，Ｖｏｌ．１５，Ｎｏ．１，Ｊａｎ．，ｐｐ．２－９（１９７４）

5）河田，天野，武田，森：ミニコンピュータを用いたカナ漢字変換システム，電子通信学会技報，ＰＲＬ７６－４２（１９７６）

6）牧野，木沢：べた書き文の分かち書きと仮名漢字変換，情報処理学会論文誌，Ｖｏｌ．２０，Ｎｏ．４，ｐｐ．３３７－３４５（１９７９）

7）牧野，木沢：べた書き文の仮名漢字変換システムとその同音語処理，情報処理学会論文誌，Ｖｏｌ．２２，Ｎｏ．１，ｐｐ．５９－６７（１９８１）

8）大河内，藤崎，諸橋：仮名漢字変換のための文法解析，計算機言語学研究会資料２５－４（１９８１）

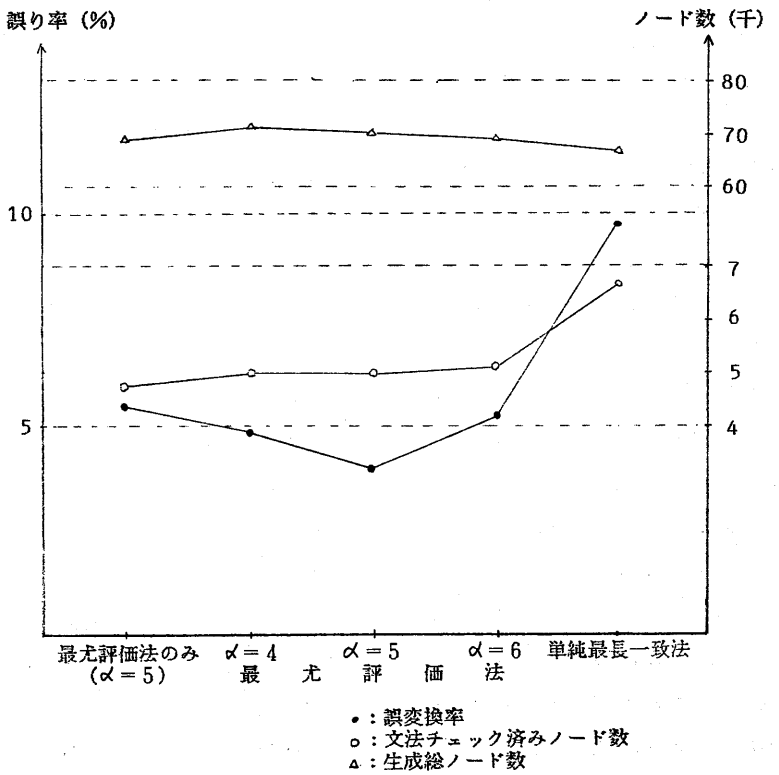

図１０　評価結果