Vol. 24 No. 4 情報処理学会論文誌 Jul. 1983

分かち書き方式仮名漢字変換のためのバックトラックを必要としない文法解析†

大河内 正明††

日本語文書処理システムの研究の一環として，仮名漢字変換方式の日本語エディタを試作したので，これを文法情報の取扱いを中心にして報告する．この仮名漢字変換の入力文は文節分かち書きを原則とするが，分かち書きを柔軟にするために，拡張した文節を導入してある．同音異義語の選択は，文法規則との整合性と単語の頻度情報に基づいて決めている．従来の文法解析は，各単語の接続条件を1対ずつ対比し，複数の接続可能性に対してはバックトラックなどの複雑な木探索を必要とするものが多いが，本システムでは仮名漢字変換は付属語連鎖の語構成を必ずしも問題にしないという特徴を積極的に利用して，接続ベクトル法と呼ぶバックトラックを必要としない簡潔で効率的な分析法を実現した．また，従来の文法解析は，自立語候補を引き当てるたびに，それに続く付属語連鎖を分析して候補の妥当性を検定するものが多いが，接続ベクトル法では，従来複雑であるとされていた文節終端からの分析も効率よく処理できるようになり，1回の付属語連鎖の分析で自立語候補の条件がすべて求まるので，その後の自立語候補の接続検定が効率的になるだけでなく，辞書にない片仮名表記語の推定や，未登録語の品詞・活用の推定も容易になっている．なお，漢字を含む拡張付属語は，構成ベクトルと呼ぶビット列の導入による2段階形態素解析で扱っている．

1. まえがき

日本語文章の入力法として，仮名で入力したものを計算機で漢字仮名交り表記に変換するいわゆる仮名漢字変換は，一般の人にとって使いやすいため，九大の栗原ら[1]の研究に始まり数多くの研究がなされてきた．

仮名漢字変換の入力形式としては，①辞書引き単位明示，②字種指定[3]，③文節分かち書き[2),3)☆]，④べた書き[4]の四つが代表的であり，ほぼこの順には打鍵数が少なくなりユーザの負担も少なくなるが，処理はむずかしくなる．仮名漢字変換の処理においては，辞書引き単位の抽出と同音異義語の選択の二つが主要課題であり，同音異義語の優先順序づけと並んで，文法との整合性の検定（形態素解析レベル）が広く利用されている．ただし，①は文法情報を使わず，用言を終止形で変換してから修正するなど操作がやや煩雑であり，④は処理がむずかしいだけでなく，あいまいさも増大しやすい（例：キノウハイシャニイッタ→昨日は医者に行った，昨日歯医者に行った）ので，実用上は②と③が着目されている．本稿では，文節分かち書き入力の仮名漢字変換を主対象として，形態素解析の改良について述べる．

文節の形態素解析の方法としては，Ⓐ自立語と付属語列の一括分離[5]，Ⓑ接続行列による接続検定[2),3)]，Ⓒ品詞・活用コードによる接続検定[6]，Ⓓオートマトン・モデルによるリスト処理[7]などがある．Ⓐは出現する全付属語列をまとめた表を用いて，文節から付属語列を一括して切り取り自立語を抽出する方法である．これは限定された文節表現だけを扱う初期の自動翻訳で広く使われたが，多様な文節表現を扱う仮名漢字変換への適用は困難である．Ⓑ〜Ⓓは接続条件の管理の仕方が異なるが，いずれも各単語間の接続条件を分析して，それらの組合せからなる多様な文節表現を扱おうとするものである．ただし，これらは接続条件を1対ずつ対比するので，複数の接続可能性を扱うときは，バックトラックなどの複雑な木探索が必要になる．なお，Ⓐでは英単語の語尾変化処理と同様の発想で文節終端から処理されているが，Ⓑ〜Ⓓでは自立語候補を引き当ててから付属語候補を順次接続検定する方法が一般的である．これは，解釈の多様性（枝分かれ数）に応じて処理の複雑さが増すⒷ〜Ⓓでは，文節終端より接続条件の強い自立語後端から分析したほうが，解釈の多様性が少なく処理しやすいためである（ただし，各自立語候補ごとに付属語連鎖の解析が必要）．

筆者らは，文書処理システム「ことだま」[10]の研究の一環として，仮名漢字変換方式のエディタを試作した．この仮名漢字変換の入力文は文節単位の分かち書

† Backtracking-free Grammatical Analysis for Phrase-based Kana-to-Kanji Conversion by MASAAKI OKOCHI (Science Institute, IBM Japan, Ltd.).

†† 日本アイ・ビー・エム(株) サイエンス・インスティチュート

☆ 文節より細かい分かち書き方式として，自立語・付属語分かち書き，単語分かち書きなどもあるが，打鍵しにくく実用上あまり着目されていない．

きを原則としているが，分かち書きを柔軟にするために，拡張した文節を導入してある．また，仮名漢字変換の文法解析は自立語候補の文法的整合性を検定することが主目的であり付属語列中の語構成を必ずしも問題にしないという特徴を積極的に利用して，接続ベクトル法と呼ぶ，バックトラックを必要としない簡潔で効率的な分析法を実現した[8),9)]．文節終端からの分析が接続ベクトル法で効率よく処理できるようになり，1回の付属語連鎖分析で自立語の端点と接続条件の可能性がすべて求まり，それから自立語候補を引き当てるため，自立語辞書のアクセスが少なくなる（とくに複合語の場合に顕著）だけでなく未登録片仮名表記語の推定や未登録語の品詞・活用の推定も容易になっている．

以下，本稿では，この文法解析の内容を従来の方法と対比しながら述べる．

2. 入力形式と拡張文節

本システムの入力文は分かち書きを原則としているが，字種指定も併用できる．分かち書き単位としては，通常の文法で定義されている文節だけでなく，次のような拡張文節（本稿ではたんに文節* と呼ぶ）が許される．

表1 自立語の分類
Table 1 Classification of content words.

記号	品詞・活用	例
5K	カ行五段活用動詞語幹	書（く）
IK	同上（特殊）	行（く）
5G	ガ行五段活用動詞語幹	泳（ぐ）
5S	サ 〃	押（す）
5T	タ 〃	立（つ）
5N	ナ 〃	死（ぬ）
5B	バ 〃	飛（ぶ）
5M	マ 〃	進（む）
5R	ラ 〃	走（る）
5W	ワア 〃	買（う）
M1	一段活用動詞不変化部（体言）	受け（る）
D1	同上（非体言）	見（る）
MS	サ変動詞（名詞型）語幹	検討（する）
DS	サ変動詞（する型）語幹	察（する）
DZ	サ変動詞（ずる型）語幹	案（ずる）
DK	カ変動詞漢字部	来（る）
KY	形容詞語幹	高（い）
KD	形容動詞語幹	静か（だ）
ME	名詞	学校
RN	連体詞	あらゆる
FK	副詞	いきなり
ST	接続詞，感動詞	しかし

* 本稿では，文節・付属語の用語を拡張文節・拡張付属語の意味で用いる．自立語も通常の文法での定義とは異なる．

[<連体詞>] [<自立語>] [<付属語>…]

ただし，<連体詞>はコノ，ソノなどのコソアド系を主とするもののみであり，<自立語>* は接続関係によって**表1**のように分類してある．また，<付属語>* は助詞・助動詞だけでなく，分かち書きが柔軟になるように拡張してある．

たとえば，「そんな危険なことをしないで下さい」という文の仮名入力は，本来の文節分かち書き

ソンナ キケンナ コトヲ シナイデ クダサイ

だけでなく，以下のいずれの分かち書きでもよい．

ソンナ キケンナコトヲ シナイデ クダサイ
ソンナ キケンナコトヲ シナイデクダサイ
ソンナキケンナコトヲシナイデクダサイ

自立語と付属語の取扱いに関しては，以下のような考慮をしてある．

① 用言に関しては，語幹のみを自立語とし，活用語尾は付属語に含めてある**．ただし，一段活用動詞は不変化部分（活用語尾の1字目を含む）を自立語とする（例：「見る」は「見」が自立語）．

② 助動詞の多くは，語幹と活用語尾を分離して，それぞれを付属語としてある（例：来る そう だ）．

③ 形式名詞，補助用言など補助的に使われる語（漢字表記も含む）は付属語に含めてある（例：会ってみる つもり は あり ません，ご返事 申し上げ ます）．

④ 汎用性の高い派生化接尾辞は付属語として扱っている（例：若 さ，書き 方，高 過ぎ る，読み にく い）．

⑤ 付属語の拡張による，付属語だけの文節も許している（例：どう し よう と し て いる の です か）．

⑥ 命令形が特殊な「下さる」や「くれる」などの動詞も，付属語として規則化してある．

⑦ 動詞（カ変とサ変を除く）の連用形は体言化することが多い（例：動き，延び）ので，自立語辞書に登録してなくても扱えるように規則化してある．ただし，一段活用動詞の場合は，連用形が体言化しないものが多い（とくに1字のもの）ので，「ミガイテ→見がいて」のような誤変換が生じないように，体言とみなせるか否かで分類して管理してある．

⑧ 五段活用動詞からは可能動詞や他動詞・使役動詞が派生できることが多い（例：動く→動ける，動かす；読む→読める，読ます）．これらの派生動詞は，自立語辞書に無くても扱えるように規則化してある．

** 活用語尾を付属語とは別に処理する形態素解析も多い[2),6)]が両者を統一的に扱ったほうが，処理が簡単になるだけでなく，補助用言も扱いやすくなる．

⑨ 慣用句や例外的な接続関係は単一の付属語としてある（例：言わざるをえない，よさそうだ）．

3. 仮名漢字変換処理の概略

本システムの仮名漢字変換の処理の概略を図1に示す．入力文に対する分析は，分かち書きの場合の空白，字種指定の場合の字種変化によって抽出した文節の単位で行う．文節は，終端から付属語連鎖を解析し，前端から連体詞を引き当てることによって，自立語の可能性をしぼってから自立語候補を検定する．自立語候補が複数個残る場合は，使用頻度等によって第1候補を選ぶ．それが正しくないときは，ユーザの指示によって他の候補と置き換える．同音異義語から一度選択された自立語は，そのセッション中は第1候補となるので，同じ同音異義語選択を繰り返すことはない．

単一自立語候補で妥当なものがない場合は複合語の可能性が調べられる．それでも妥当なものが見つからない場合は，字種指定があれば，接続条件を無視して字種指定に合うものを探す．字種指定がなければ，辞書にない片仮名表記語が使われていると仮定して，文脈からこれを推定して片仮名表示する（これは適当な変換候補が無かったことの警告をも兼ねている）．

自立語の辞書としては，全ユーザに共用される基本辞書とユーザごとの使用傾向を反映するユーザ辞書がある．基本辞書にない自立語でも，当て字などの校正処理で正しく変換されると，文脈から推定される品詞・活用情報とともにユーザ辞書に登録される．

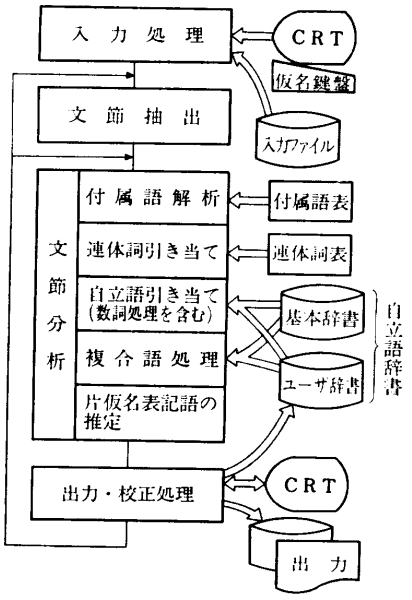

図 1 仮名漢字変換処理の流れ
Fig. 1 Flow of Kana-to-Kanji conversion.

4. 文法情報の表現と管理

4.1 リンクによる接続条件の表現

単語間の接続条件はリンクという概念でまとめてある．各単語（同形多義語をまとめて1単語とする）は前後の接続条件に対応して，前端リンクと後端リンクの対を何組かもち，二つの単語は共通するリンクを介して接続可能である．また，単語の一端で境界条件（名詞に後続するとか文節の最後になるなど）を与えてリンク群を制限すると，他端で許されるリンク群も

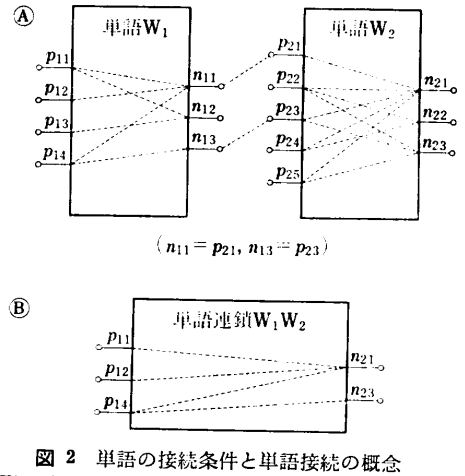

図 2 単語の接続条件と単語接続の概念
Fig. 2 Conceptual view of word connection conditions.

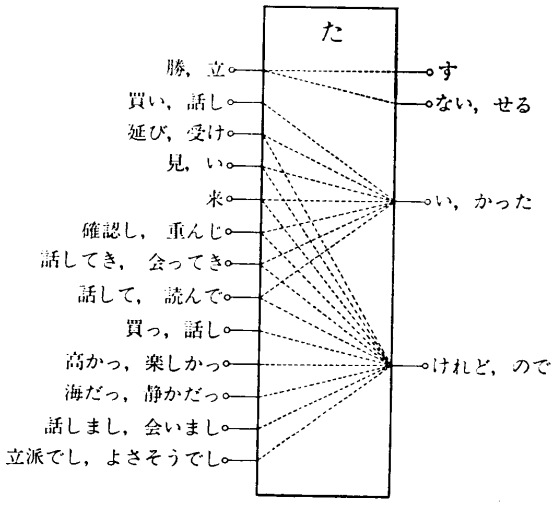

図 3 付属語「た」の接続条件と接続例
Fig. 3 Connection condition of function word "ta."

変わる．たとえば，**図2**Ⓐは単語 W_1 に (p_{11}, n_{11}) などリンク対が6対，単語 W_2 に同様に10対ある場合を示している．ここで，両者に共通リンクが2組 $(n_{11}=p_{21}, n_{13}=p_{23})$ あると，両単語は接続して単語連鎖全体として図2Ⓑの接続条件をもった単一単語と同等に扱うことができる．また，W_2 の後端に n_{23} しか許されなければ，W_1W_2 の前端では p_{14} しか許されない．

リンクの設定の仕方はいろいろ可能だが，本システムでは付属語連鎖を文節終端から逆方向に分析するため，後端リンクが少ないほうが処理しやすいので，後端リンクが文法概念と対応して単純になるように設定してある．たとえば付属語「た」は，**図3**のように表現してある．なお，リンクの総数は，現システムでは96種（内22種は表1の各自立語に対応）である．

4.2 接続ベクトルの導入

複数リンクからなる接続条件を効率よく処理するために接続ベクトルというビット列を導入してある．接続ベクトルは各ビットの1/0が各リンクの有/無に対応して**図4**のように構成されている．そのうち自立語の後端リンクに対応する部分だけを自立語ベクトルと呼ぶ．また，付属語連鎖の分析の制御に使われる制御フラグ* も特殊なリンクとして含まれている．

接続ベクトルによる接続条件の分析（接続ベクトル法）の最大の特徴は，語構成を問題にせず，同一文字

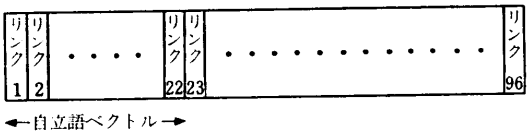

図4 接続ベクトルのビット構成
Fig. 4 Bit allocation in a connection vector.

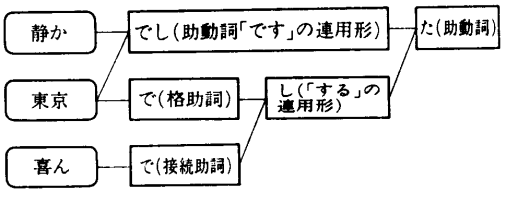

図5 付属語連鎖「でした」の語構成解釈
Fig. 5 Decomposition of "deshita" into function words.

* 制御フラグには，すべての付属語の前端リンクに含まれるFフラグと，文節頭にも使われる可能性のある付属語（補助用言など）の前端に含まれるSフラグがある．Fフラグは付属語連鎖の分析中，付属語を引き当てる次の位置（次に長い付属語連鎖の前端）を探すのに用いられ，Sフラグは文節前端にある場合，自立語の引き当てを省けることを示す．

逆引き仮名見出し (FZK)	漢字部インデクス (KJX)	後端リンク (NXT)	前端接続ベクトル (PVTR)

図6 付属語表の形式
Fig. 6 Format of function word table.

列の単語連鎖の接続条件をまとめて扱え，バックトラックが不要な点である．たとえば，文節の最後にくる「でした」の語構成は，文法的には**図5**のように3通りあり，文節終端から分析する場合，単語を1対ずつ接続検定する方法では，「でした」の前の候補単語は3通りの「でした」と別々に接続検定しなければならないが，接続ベクトル法では，3通りの「でした」の接続条件の和集合が接続ベクトルでまとめて表現されるので，「でした」の前の候補単語との接続検定は，「でした」が単一単語であるかのように処理できる（両者の接続ベクトルの論理積をとって1のビットが残れば接続する）．語構成の解釈が何通りあっても，各文字位置ごとに一つの接続ベクトルを用意して，単語連鎖の長さの順に処理すればよく，バックトラックは不要である．しかも主要な処理はビット列の論理演算であり，高速で，ファームウェア化にも適している．なお，後出の図8等で例示するように，分析の中間結果として得られる単語連鎖の端点の接続条件も，すべて統一的に扱える．

4.3 自立語辞書と付属語表

自立語辞書は自立語ベクトルを頻度情報などとともに管理している．自立語ベクトルは，多品詞語** に対しても複数ビットが1になるだけで統一的に扱えるので，品詞・活用コード[6]などよりも汎用で扱いやすい．

付属語とその接続条件は，**図6**の付属語表（主記憶域上では木構造で構成）で管理してある．付属語連鎖を文節終端から分析しやすいように，見出しは逆引きになっており，接続条件も各後端リンクごとに対応する前端リンク群を接続ベクトル表現にしてある．漢字を含む付属語の管理と処理は後出の図9で例示する．

5. 文法情報の処理

本システムの形態素解析は，

① 単語連鎖の端点の接続条件の分析

② 構成単語固有の情報の抽出

の2段階に分けて処理している．漢字を含む付属語を

** たとえば，「満足」は名詞，形容動詞語幹，名詞型サ変動詞語幹になる多品詞語である．

扱う場合など，構成単語を識別してその固有の情報（漢字表記）を使う場合は，②の処理が必要になるが，ほとんどの文節は①の処理だけでよい．①は接続ベクトル，②は構成ベクトルと呼ぶビット列がそれぞれ中心的役割を果たしている．

5.1 作業域とその初期処理

付属語連鎖の分析は図7の作業域上で行う．接続ベクトル $\boldsymbol{a}_i$ は，文字列 $I_iI_{i+1}\cdots I_L$（L は文節長）が付属語連鎖の場合のその前端リンク群に対応する．また，構成ベクトル $\boldsymbol{s}_i$ は，この付属語連鎖を構成している付属語を示すビット列であり，この第 m ビットが1ならば，構成付属語表* の第 m エントリーに対応する付属語が，この付属語連鎖の要素であることを示す．漢字フラグ f_i は，$\boldsymbol{s}_i$ で示される付属語群中に漢字を含む付属語があることを示す1ビットである．

主要な処理は，$\boldsymbol{a}_{L+1}$ を文節終端接続ベクトル $\boldsymbol{e}$** に設定し，他のすべての $\boldsymbol{a}_i, \boldsymbol{s}_i, f_i$ のビットを0にしておいて，以下のように分析することである．

5.2 付属語の引き当て

文字例 $I_{i-l}I_{i-l+1}\cdots I_{i-1}$ が付属語候補として見つかったとき，その（後端リンク，前端接続ベクトル）の対として $(n, \boldsymbol{p})$ があり，接続ベクトル $\boldsymbol{a}_i$ の第 n ビットが1ならば，この付属語候補を採用し，$\boldsymbol{a}_{i-l}$ を

$$\boldsymbol{a}_{i-l} \leftarrow \boldsymbol{a}_{i-l} \vee \boldsymbol{p} \qquad \text{（ビット列の論理積演算）}$$

と更新する．この付属語が第 m 番目の採用付属語ならば，構成付属語表の第 m エントリを追加するとともに，構成ベクトル $\boldsymbol{s}_{i-l}$ を，

	1	2	⋯	i	⋯	L	$L+1$	
KANA	I_1	I_2	⋯	I_i	⋯	I_L		
AVTR	$\boldsymbol{a}_1$	$\boldsymbol{a}_2$	⋯	$\boldsymbol{a}_i$	⋯	$\boldsymbol{a}_L$	$\boldsymbol{a}_{L+1}$	
SVTR	$\boldsymbol{s}_1$	$\boldsymbol{s}_2$	⋯	$\boldsymbol{s}_i$	⋯	$\boldsymbol{s}_L$	$\boldsymbol{s}_{L+1}$	
KFLG	f_1	f_2	⋯	f_i	⋯	f_L	f_{L+1}	

I_i：入力仮名文節の i 文字目
$\boldsymbol{a}_i$：付属語連鎖 $I_iI_{i+1}\cdots I_L$ の前端の接続ベクトル
$\boldsymbol{s}_i$：〃 の構成ベクトル
f_i：〃 の漢字フラグ

図7 文節分析作業域
Fig. 7 Work table for phrase analysis.

$$\boldsymbol{s}_{i-l} \leftarrow \boldsymbol{s}_{i-l} \vee \boldsymbol{s}_i$$

と更新した上で，第 m ビットを1にする．また，この付属語が漢字を含む場合は漢字フラグを $f_{i-l}=1$ とし，含まなければ $f_{i-l}=f_i$ とする（後出の図9参照）．

5.3 自立語の接続検定

文字列 $I_1I_2\cdots I_{j-1}$ が自立語ベクトル $\boldsymbol{b}$ をもつ自立語候補のとき，接続ベクトル $\boldsymbol{a}_j$ 内の自立語ベクトル $\boldsymbol{a}'_j$ に対して，ビット列の論理積をとり，

$$\boldsymbol{b} \wedge \boldsymbol{a}'_j \neq \boldsymbol{0}$$

ならば，この自立語候補は，後続付属語連鎖と整合するとして採用する．なお，$\boldsymbol{a}'_j=\boldsymbol{0}$ の場合は，文字列 $I_1I_2\cdots I_{j-1}$ を自立語辞書から検索する必要もない．

5.4 漢字を含む付属語の処理

自立語 $I_1I_2\cdots I_{j-1}$ と整合した付属語連鎖が漢字を含む場合（$f_j=1$ でわかる）は，構成ベクトル $\boldsymbol{s}_j$ によって，漢字を含む付属語を探し出す（後出の図9参照）．ただし，$\boldsymbol{s}_j$ で示される語構成に複数の可能性があり，漢字を含む付属語の位置全体が他の付属語群と重なっている場合には，漢字表記するか否かが一意に決まらないが，この場合でも付属語連鎖の左端に境界条件（自立語ベクトル）を与えて，順方向に接続条件を確認すれば，実際の構成付属語がわかる．

6. 文節分析例

6.1 単一自立語の文節の分析

入力文節が「キケンシナイデクダサイ」の場合の処理概念を図8に示す．まず，文節終端の接続条件（文節終端接続ベクトル）を設定して，そこから逆方向に文節の最後になりうる付属語をすべて求める．この例では，「い」，「際」，「下さい」の三つが求まり，その前端の接続条件（接続ベクトル）が算出される．以後，分析済みの付属語連鎖の短いものから順に，その前に接続しうるすべての付属語を求めていく．その結果，付属語連鎖としては，最短の「い」から最長の「しないで下さい」までの八つが求まり，これらの前端と文節終端に接続条件が残る（図中には自立語ベクトル分だけを表1の記号で示してある）．これら9位置のうち，自立語と接続可能な8位置が自立語の後端候補になる．この例では連体詞は引き当てられず，自立語の前端候補は文節前端のみである．両端位置の候補を満たす自立語としては，「危険」と「棄権」が見つかっているが，後者のみが名詞型サ変動詞語幹として接続条件を満たし，出力文節「棄権しないで下さい」

---

* 構成付属語表は，後出の図9に例示してあるように，付属語の両端位置と付属語番号の表である．

** 文節終端接続ベクトル $\boldsymbol{e}$ は，文節の最後になりうるリンク群に対応して設定するが，後続文節前端にSフラグがある場合は，まずその文節前端の接続ベクトルを $\boldsymbol{e}$ として第1変換候補を求める．したがって，「キケンスル」を区切って「キケン スル」と入力しても同じ結果「棄権する」を得る（区切ると「キケン→危険」や「スル→刷る」などの同音異義語の可能性も生じるが，これらは第2変換候補以下として扱われる）．

が得られる．以上の処理における作業域の使われ方と漢字を含む付属語の取扱いを**図9**に示す．

入力文節が「カワセマセンデシタ」の場合は**図10**のようになる．自立語端点候補を満たす自立語候補は，

為替，川，乾(く)，蚊，買(う)，書(く)

など20以上あるが，接続条件を満たすものは

買(う)，飼(う)，交(す)，かわ(す)

の四つだけであり，これらのなかで「買(う)」が最も使用頻度が高いとき，「買わせませんでした」が第1変換候補になる*．

従来のように，自立語を引き当ててから後続する付属語連鎖を分析する方式の場合は，自立語候補（字種指定が無いと，図8の場合，「木」や「聞(く)」なども自立語候補になる）のそれぞれに対して付属語連鎖を分析しなければならないが，本方式では1回の付属語連鎖分析で自立語の端点候補と接続条件のすべてが求まるので，端点位置候補を満たす自立語候補だけを，付属語連鎖の接続条件と対比するだけでよい．

6.2 複合語を含む文節の分析

本システムでは付属語連鎖を先に分析しているため，複合語の処理も効率的になっている．たとえば入力文節が「カクリツブンプニハ」の場合は，**図11**のように付属語連鎖は最長のものでも2文字であり，それに達する単一自立語候補がないことからただちに複合語処理に入っている（従来の自立語を先に引き当てる方式の場合は，字種指定がないと，「確率，確立，隔離，核，蚊，書(く)，買(う)」など，20以上の単一自立語候補の接続検定がすべて失敗してから初めて複合語処理に入る）．語基（複合語構成要素）の引き当て**も，複合語の後端候補が与えられているため，字種指定がある場合に近い効率で処理できる．

6.3 未登録片仮名表記語の推定

外国固有名詞や外来語などの片仮名表記語は，数が多く表記のゆれも大きいため，前もっ

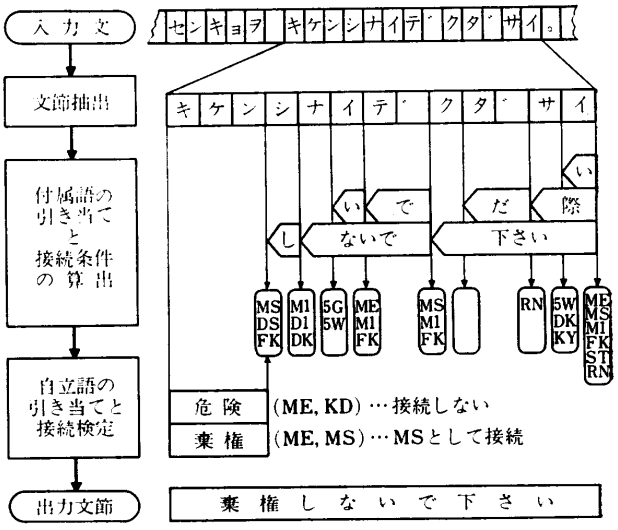

図8 文節分析の処理概念
Fig. 8 Conceptual view of phrase analysis.

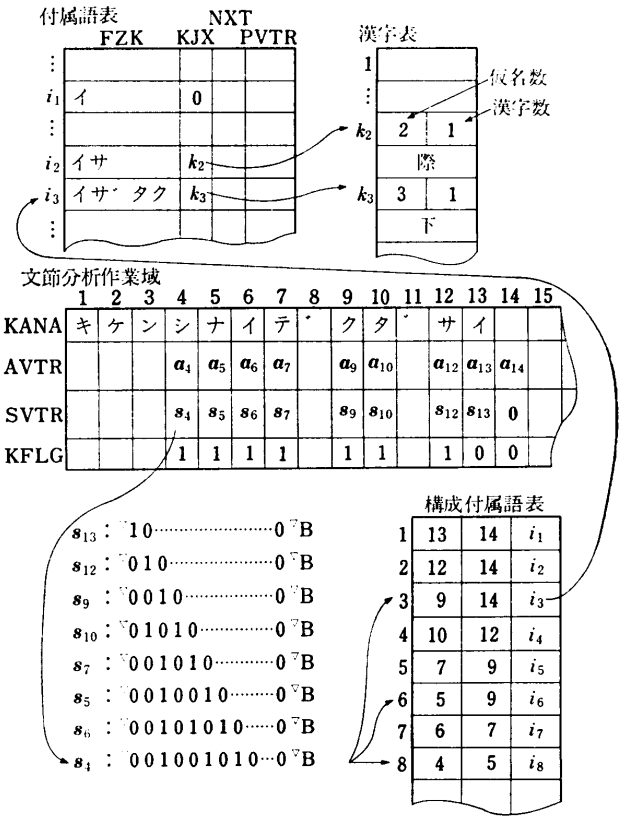

図9 付属語連鎖の構成情報の関連
Fig. 9 Relation of constituent information of function word string.

* この例で，漢字表記の「**綴**(す)」が辞書に無い場合でも，単漢字の引き当てによって，「綴せませんでした」とすれば「**綴**(す)」はサ行五段活用動詞語幹と推定されて，ユーザ辞書に登録される．

** 語基の引き当ては最長一致を優先する方式がとられることが多いが，本システムでは処理効率を上げるため，漢字の読みの性質を使って語基推定している[11]．

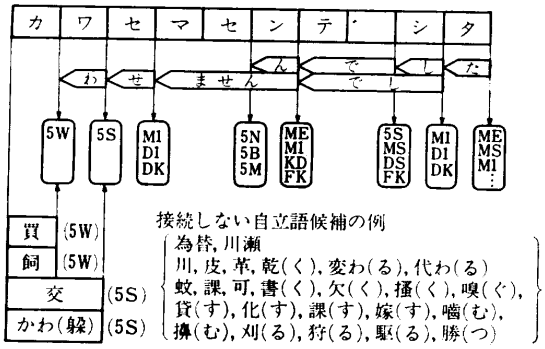

図 10 同音異義語の多い文節の分析
Fig. 10 Analysis of a phrase including many homonyms.

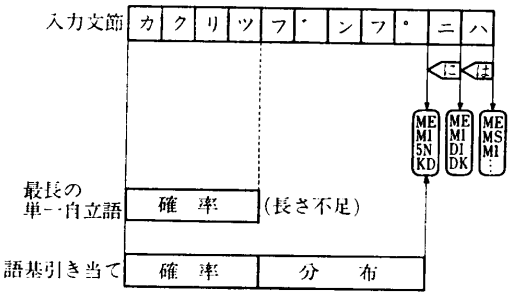

図 11 複合語を含む文節の分析
Fig. 11 Analysis of a phrase including a compound word.

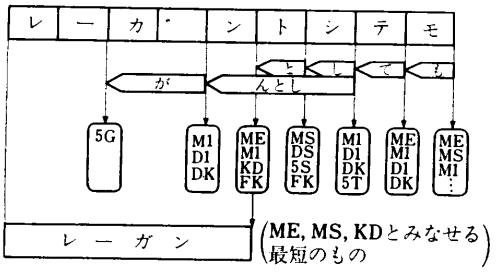

図 12 辞書にない片仮名表記語の推定
Fig. 12 Estimation of an unregistered loanword.

てすべて辞書に用意することは困難である．片仮名表記語は，名詞（例：パリ），名詞型サ変動詞語幹（例：カットする），形容詞語幹（例：フレッシュな）として使われることが多いので，複合語処理によっても適当な変換候補が見つからず字種指定もないときは，辞書にない片仮名表記語が使われていると仮定して，これを図 12 のように文脈から推定している．

7. むすび

試作した仮名漢字変換システムを文法情報の取扱いを中心に報告した．おもな特徴を以下にまとめて述べる．

① 形式名詞・補助用言を付属語扱いにすることによる文節の扱張は従来から行われているが，本システムではこれをさらに進めて，付属語の大幅な拡張，自立語に前接する連体詞の処理，派生化規則の導入などにより，分かち書きを柔軟にするとともに自立語辞書項目を少なくしている．

② 単語連鎖の端点の接続条件を，語構成を問題にせず接続ベクトルで統一的に表現しているため，複数の接続可能性がバックトラックなしに，ビット列の論理演算を主とする簡潔な処理で効率よく分析できる．これは複数の接続可能性を同じ深さ（同一文字列）に関して接続ベクトルでまとめて求める有限非決定オートマトンに相当している．等価な有限決定オートマトンで実現した場合よりも，状態（リンク）の数が少なくなるだけでなく，必要記憶域も少ない．付属語約 250（文字列としての異なり数），連体詞約 20 を漢字表記も含めて管理するのに約 6k バイト，最長 30 字の文節の付属語連鎖の分析の作業域に約 500 バイト必要とするだけである．

③ 文節分析は，文節終端からの 1 回の付属語連鎖分析で，自立語の端点と接続条件の可能性をすべて求めてから自立語候補を引き当てるため，自立語候補へのアクセスが少なくなる（とくに複合語の場合に顕著）だけでなく，未登録片仮名表記語の推定や未登録語の品詞・活用の推定も容易になっている．

④ 構成ベクトルを接続ベクトルと組み合わせた 2 段階形態素解析は，単語連鎖の構成単語を扱えるので，一般の形態素解析（順方向でもよい）にも適用できる．単語の抽出がむずかしく同音異義語の多い日本語文を，文字列長に比例する程度の処理量で効率よく分析できる．

本稿の内容は実験システムで実働化（ただし漢字を含む付属語の処理は簡略化した方式[8]で実現）して実証されているが，これを基礎にした IBM 日本語文書処理システム[12]では若干機能が異なっている．なお，以下の点の改善の検討も進めている．

① 名詞を接辞との接続関係によってさらにに分類して，複合語の変換精度を上げる．

② 形容詞を派生化規則（例：高い→高まる）の適用可能性によって分類して，自立語辞書項目を減らす．

③ 「行(く)」と「(来る)」の二つは，同様の接続関係の自立語が他になく，漢字の読みも活用形で変わ

るため，付属語として扱い誤変換（例：コル→来る）を避けるとともに自立語ベクトルのビットを節約する．

④ 漢字を含む付属語に対し，漢字表記するか否かの優先度を管理して，表記のゆらぎに対処しやすくする．

本稿で述べられなかった本システムの人間工学的側面や変換精度の評価等については文献 11) に述べてある（精度等はその後改善されている）．また，文法規則[13]の内容についてはあらためて報告する予定である．

謝辞 「ことだま」プロジェクトを構成して協力いただいた藤崎哲之助・諸橋正幸の両氏，辞書の整備等に協力いただいた大深悦子・戸沢義夫・間下浩之の各氏に感謝します．なお，辞書作成には，三省堂のご厚意により，新明解国語辞典を利用させていただいた．

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，九州大工学集報，Vol. 39, No. 4, pp. 659-664 (1967).
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) 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，PRL 76-47 (1976).
4) 牧野，木澤：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).
5) 坂井，杉田，渡辺：電子計算機による和文英訳，情報処理学会計算言語学研資料 CL 69-1 (1969).
6) 長尾，辻井他：計算機による日本語文章の理解に関する研究，文部省科研費特研（1）報告書 (1979).
7) 水谷：リスト処理による活用アクセプタ，計量国語学，No. 56, pp. 6-29 (1971).
8) 大河内，藤崎，諸橋，戸沢：仮名漢字変換のための文法情報の管理と処理，IBM TSC レポート，N：G 318-1510 (1979).
9) 大河内，藤崎，諸橋：仮名漢字変換のための文法解析，情報処理学会計算言語学研資料 25-4 (1981).
10) 藤崎，大河内，諸橋，戸沢：日本語文書処理システム「ことだま」，IBM TSC レポート，N：G 318-1512 (1980).
11) 藤崎，大河内，諸橋：「ことだま」文書処理システムの文節分かち書き仮名漢字変換，情報処理学会論文誌，Vol. 23, No. 1, pp. 1-8 (1982).
12) 文書入力編集プログラム―カナ鍵盤用―プログラム解説書，IBM マニュアル N：SH 18-0037 (1981).
13) 大河内：仮名漢字変換のための形態素接続規則―作成方針と管理の概要―，IBM TSC レポート，N：G 318-1560 (1981).

(昭和 57 年 7 月 2 日受付)
(昭和 57 年12月 6 日採録)