自然言語処理 112－10
（1996. 3. 15）

日本語定型表現の分析と効率的照合アルゴリズム

安藤一秋，辻孝子，獅々堀正幹，青江順一
徳島大学工学部知能情報工学科

日本語の定型表現は，機械翻訳における正しい訳語の決定などに有用であり，その抽出や利用方法が研究されている．このほかにも，キーワード抽出における重要語の決定，文書校正，文書検索などにも，より広い意味での定型表現が利用されており，一般的な自然言語処理システムへの定型表現の分析と統一的な利用方法の実現が必要不可欠である．本研究では，各応用分野で必要とされる定型表現を広く捉えて分析し，定型表現に対する規則の形式的定義を提案する．これらの規則集合から実行形式の照合解析表を作成するコンパイラと，解析結果の入力から解析表を駆動して，規則を高速に照合するアルゴリズムを提案する．提案手法により，解析結果から利用したい多種多彩な情報を抽出して，定型表現の規則を定義するだけで，独自の定型表現の検出が行えるので，解析エンジン汎用性と定型表現の独立性が維持でき，開発と管理が効率的となる．

Analysis of Japanese Frozen Patterns and an Efficient Pattern Matching Algorithm

Kazuaki ANDO, Takako TSUJI, Masami SHISHIBORI
and Jun-ichi AOE

Dept. of Information Science and Intelligent Systems,Faculty of Engineering,
The University of Tokushima
Minami Josanjima-Cho 2-1. Tokushima-Shi 770, Japan

Abstract

Japanese Frozen Patterns are useful for the correct interpretation of machine translation systems and for making the complexity of case-structures simple. By relaxing the concept of the Japanese frozen patterns, the application can include wider natural language processing systems such as proofreading assistance, text retrieval and so on. This paper formalizes the definition of grammar rules for frozen patterns and presents an efficient pattern matching module for them. The grammar description and the pattern matching machine presented have the following desirable features: (1) The input consists of a stream of sets containing multiple components, (2) The separated components can be detected, and (3) The rules can be described by an exclusive set. A number of rules are built and the presented technique are evaluated by the simulation results for 7,500 Japanese sentences.

1．　まえがき

複数の形態素列が一つの意味として解釈できる日本語の定型表現は，機械翻訳における正しい訳語の決定[4]に有用であり，また動詞を主体とする活用語を含む定型表現をまとめて処理することは，活用語に関係する統語や格構造解析を省略できるので，解析の効率化の点でも有効である[6]．その抽出や利用方法が研究されている[6]．このほかにも，キーワード抽出における重要語の決定にも，より広い意味での定型表現が利用されている．さらに，文の短縮処理[8]，文書校正[10]（曖昧点の指摘），文書検索など自然言語処理システムへも定型表現の分析が必要不可欠である．しかしながら，形態素解析結果から定型表現を利用する場合，個々の応用分野で定型表現の規則を定義し，それら規則を照合すると，形態素解析エンジンや形態素解析辞書が応用分野に特化されたものとなり，形態素解析モジュールの汎用性が失われてしまうばかりでなく，システム全体の管理，拡張性をも圧迫する．この問題は，構文や意味解析モジュールに対しても同様である．

本研究では，形態素解析エンジンの汎用性を守るために，現在の応用分野で必要とされる定型表現を広く捉えて分析し，次の形式化をまず提案する．

(1)　定型表現に対する文法規則の定義．
(2)　文法規則に対して，動作規則の定義．

以上に加えて，これらの規則集合から実行形式の照合解析表を作成するコンパイラと，解析結果の入力から解析表を駆動して，規則の高速照合手法を提案する．

提案手法により，解析モジュールの汎用性は保たれ，各応用分野では独自の定型表現と動作を定義するだけとなるので，開発と管理が極めて効率的となる．

2．定型表現の分析と照合情報

図1に自然言語処理システムの構成を示す．前述のように，図1の外枠内で全体システムを構成すると，解析エンジンや辞書は解析結果から情報を抽出・検索するエンジンと統合化され，解析知識と分野別の固有知識も混合される．本章では，このエンジンを形態素解析として捉え，種々の定型表現の利用方法を分析・分類し，規則の定義，照合方法への準備とする．

以上より，図2に示すように，各アプリケーションに対して定型表現を定義して利用できるので，解析エンジンの汎用性は保たれる．

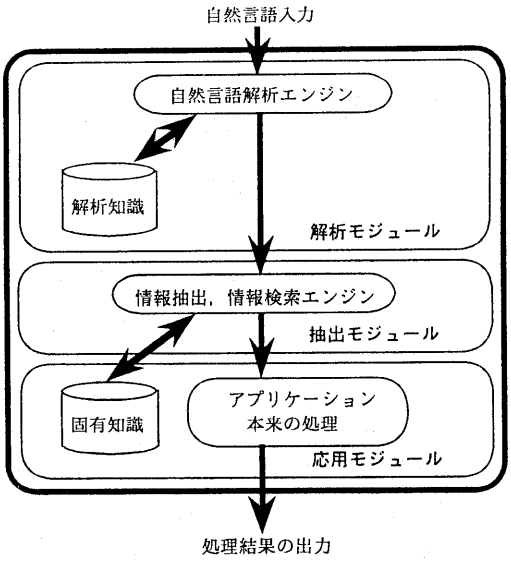


図1　自然言語処理アプリケーションの構成

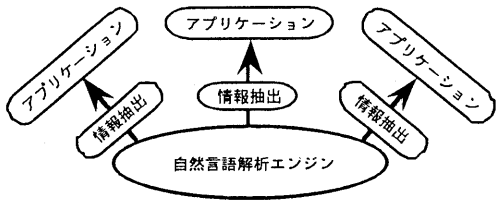


図2　汎用的自然言語エンジン

2．1　定型表現の利用形態

（A）助詞相当語句（関係表現）

“に関して”のように助詞相当の働きをもち，語の挿入や交換が一般的に行えない慣用的な表現の一種であるので，一語にまとめて解釈するのが有効である．機械翻訳における適切な意味解釈，文の短縮，格構造析の効率化など非常に多くの分野[6]～[9]で利用できる．しかし，応用分野により取り扱う関係表現の分類は異なるので，その表現（助詞＋用言＋付属語）の範囲が曖昧である．例えば，機械翻訳では意味解釈が一意の関係表現を利用すべき制約があるが，利用者への判断を促すような提示をする文章校正や文短縮では，関係表現の定義も異なる．

下線で示す形態素列の置換候補を次の（）内に記述した例を考える[9]．

(1)　言語に関する（の）処理を行う．
(2)　木で作った（の）椅子に座る．
(3)　服に付着している（の）埃をはらう．
(4)　法隆寺に続いている（への）道を進む．

(5) 外国に留学している（の）友人に聞く．

文例(1)の定型表現「に関する」の助詞「の」への短縮は，文解釈に影響を与えないと思われるが，文例(5)の置換処理では，「留学している」が失われるので，元の文の意味を正しく解釈するのは，文脈を考慮しないと難しくなる．また，文例(2)～(4)の短縮処理で失われた意味は，定型表現に関係する名詞と文の主動詞から推論可能と思われる．また，名詞間の助詞の意味が曖昧な場合，文書校正[10]では，短縮とは逆に関係表現の記述を提示しなくてはならない．

（B）助述表現

述部の付属語の定型表現[7]と考えられる表現であり，(A)の関係表現と同様にまとめて意味解釈した方が有効である．例えば，“する必要がある”，“ねばならなかった”などであり，この分類も多彩である．

（C）慣用的表現

語結合を構成する要素の意味が逸脱を起こす表現であり，特に述語慣用句（“心が沈む”，“油を売る”など）と機能動詞表現（調査を行う）は，

（名詞＋助詞）＋（用言＋付属語）

の構造をもっている．しかし，構造間に他の要素（副詞，各要素など）が挿入可能なもの，名詞要素を修飾する要素制約や意味の考慮など，単なる構造の並びだけでは判定できない定型表現と考えることができる．

（D）書き替え表現

書き替えることで，構文解析などの上位の解析上の曖昧性が少なくなる場合[5]（縮約展開型の書き替え：私は東京，彼は大阪を担当する＝＞東京を担当し，）；翻訳上，無意味である冗長語を書き替え留場合（調査するものである＝＞調査する）；文章校正[10]での同義語の提示（パソコン＝＞パーソナルコンピュータ）；同じ意味で文字列の短い表現への書き換え[8]（しかしながら＝＞しかし）など，書き替え規則を定義したい応用分野は非常に多く，その定義も多彩である．

（E）係り受け関係

“高速なコンピュータの処理”のように係り受けの曖昧性の検出は，局所的な探索で決定できる定型表現と考えられ，解析の曖昧性[11]の検出や文書校正での曖昧な表現の指摘対象となる．

（F）誤り表現，不適切な表現の検出

重なり語の訂正（本を読書する＝＞本を読む）；副詞の呼応の検出と訂正（決して泳ぐ＝＞決して泳がない）；助詞の重なりの検出（文書ののの校正）；及び文語，敬語，会話調の検出など定型表現として定義できる．この処理が十分でないと，根本的な意味解釈は不可能となるので，日本語解析において，最も重要な位置を占めると考えられる．

（G）重要語の抽出

自立語による造語に基づく複合語[11]は，多数存在し，文書を要約する重要語（キーワード），文書データベースでの検索語の抽出に関係する．特に，キーワードの決定では，複合語“自然言語処理”を短い単位で抽出するのか，長い単位で抽出するのか，あるいは，キーワードの重み付け（評価値の加減算）も利用分野によって異なる．例えば，接頭辞＋名詞（“各”＋“手法”）のキーワードでは，前の形態素を除く処理や，「AのB」のAを削除するが，Bは用言性名詞でないなど多彩な定型表現の定義と応用が考えられる．

（H）文と文の関係，文書内での関係

以上は，文内処理が中心であったが，文の関係を定型表現として，定義することも可能である．例えば，文S1,S2が，“．．ので”，“ために”の＜理由＞を表現する文である場合，文S1,S2の一つの解釈関係として，＜理由＞＋＜理由＞なる定型表現を定義できる．

この考え方は，段落単位でも同様である．また，手紙文における，“前略”＋“草々”や，論文記述における“まえがき”＋“むすび”の関係も，その情報が抽出できるならば，文書全体の定型表現として定義し，検出・検査することが可能となる．

2．2 定型表現の規則

定型表現の規則には，形態素の字面（表記）や品詞情報に加えて，形態素の意味属性や特別な辞書参照などの記述が必要となる．また，形態素の表記は非常に項目数が多いので，合理的に抽象化しないと規則数の爆発を招くことになる．従って，形態素で考慮すべき多属性情報（表記，品詞，意味属性など）を含めた効率的な規則定義と，その規則に適合した照合エンジンの提案が必要である．従って，多属性項目の照合では，複数項目の部分一致のために，集合の包含関係の演算を導入する．

以後の議論で重要なことは，定型表現が形態素情報にとどまらないことである．即ち，解析結果から抽出される情報であれば，文節単位，文単位，段落単位等どのようなものであっても規則に反映できるので，以後，形態素は単に構造と呼ぶ．

【集合表現による構造の定義】

構造Mは，属性attrとその値valueからなる組(attr,value)を要素とする集合とする．但し，形態素を構造とする場合は，必須要素である表記と品詞の属性名をそれぞれSTR(string)とCAT(category)で

表し，その他の属性名は自由に設定できるものとする．また，簡単のために活用情報は品詞情報に含めるものとする．

例えば，“医者”の形態素構造Mは，次となる．
{(STR,"医者"),(CAT, Noun), (ATTR1,Hum)}

【属性値のアクセス関数の定義】

この意味属性ATTR1とその値Hum（人間の意味）が任意要素である．なお，形態素構造Mに対する属性値は，次の関数fで得られるものとする．

f(M, 属性名)=属性値

上記のMでは，

f(M,STR)= “医者”
f(M,CAT)=Noun
f(M,ATTR1)=Hum

となる．

【構造連接演算子の定義】

定型表現の構造列は，かならず連続するとは限らないので，構造が連続するか否かを示す2項演算子Bを導入する．即ち，"M+M'"がMとM'が連続することを表し，"M*M'"がMとM'と間に0個以上の構造が存在することを表す．'*'をもつ規則を特に**分離規則**と呼ぶ．規則M+M，M*Mに対しては，両者が検出されるものとする．

［**定義1**］p番目の規則Rule(p)を次のように定義する．

$Rule(p)=B_{p1}M_{p1}B_{p2}M_{p2}...B_{pnp}M_{pnp}$（$1 \leqq np$）

但し，B_{p1}は，'+'に限定される．

解析の結果から抽出された構造Nで表して，規則の構造Mと区別し，構造の照合は，集合の包含関係（NがMを包含する）が成立するか否かで行われる．

【辞書のアクセス関数の定義】

構造Nに対する種々の任意属性値を解析辞書に登録するのは，本研究の目的に反する．従って，形態素構造Nを例とすると，必須要素（STRとCAT）と規則に記述された属性ATTRからATTRの属性値を得る関数Attr_Searchを次のように定義する．

Attr_Search(f(N,STR), f(N,CAT), ATTR)

引数は，形態素の表記f(M,STR)，品詞f(M,CAT)，属性分類ATTRである．

上記の“医者”の形態素構造では，ATTR1を別の辞書に定義しておくことで，次が得られる．

Attr_Search(“医者”, Noun, ATTR1)=Hum

2．3　定型表現の規則の例

集合表現を導入することでより多彩な照合が可能となる．例えば，“名詞＋の＋名詞”を記述する次のp番目の規則を考える．

$Rule(p)=B_{p1}M_{p1}B_{p2}M_{p2}B_{p3}M_{p3}$

M_{p1}={(CAT, Noun)}
M_{p2}={(STR, “の”), (CAT, Post_P[注1])}
M_{p3}={(CAT, Noun)}
$B_{p1}=B_{p2}=B_{p3}$='+'

このRule(p)は，次の形態素構造列と照合できる．

N_1={(STR, “母”), (CAT, Noun)}
N_2={(STR, “の”), (CAT, Post_P)}
N_3={(STR, “絵”), (CAT, Noun)}

また，Rule(p)の演算子をB_{p3}='*'と変更することで，次の形態素構造列と照合できる．.

N_1={(STR, “母”), (CAT, Noun)}
N_2={(STR, “の”), (CAT, Post_P)}
N_3={(STR, “美しい”), (CAT, Adj)}
N_4={(STR, “絵”), (CAT, Noun)}

さらに，”動物の絵”などは除外して，

“＜人間（意味属性）＞の絵”

と制約したい場合は，次の規則を作成すればよい．

M_{p1}={(CAT, Noun), (ATTR1, Hum)}

このように，適宜属性を付加することで，表記，品詞，意味属性へと，情報の抽象化が導入できる．

3．照合アルゴリズム

定型規則の照合機械PMM(pattern matching machine)は，構造列に対して規則集合に属する規則Rule(p)に照合する位置を検出する．但し，ここで検出された構造の部分列は互いに重複することも許す．これらの特徴は，複数キーワードに対する文字列照合機械Aho & Corasic[1],[8]やトライ検索[2],[3]に類似しているが，次の拡張点をもつ．

（拡張1）　多属性項目の照合．
（拡張2）　分離規則の照合．
（拡張3）　補集合の照合．

入力構造をIとするとき，状態番号をマシンPMMの動作は次の2つの関数で制御される．

goto関数：$S \times \{*,+\} \times I \rightarrow S \cup \{fail\}$
output関数：$S \rightarrow \{p \mid$ 規則番号$p\}$

まず，拡張1に対して，goto関数の入力構造を集合として捉え，拡張2を実現するために，構造の連結演算子も遷移情報に導入する．即ち，演算子'*'につながる構造M={(CAT, Noun), (ATTR1, Hum)}に対して，

goto(s, '*', M)=s'

が定義されていれば，状態sからs'への遷移が存在する．従って，入力構造Nに対して，

$M \subseteq N$

ならば，状態sからs'への遷移を可能とする．例えば，次の構造N1は遷移でき，N2は遷移できない．
N1={(STR, “母”, (CAT, Noun), (ATTR1, Hum),

[注1] 後置詞(Postposition)の略．

N2={(STR, “犬”, (CAT, Noun), (ATTR1, Animal)}

次に，拡張２に対して，照合処理中に一時的に抽出規則候補を格納するマシンCDM(CanDidate Machine)を構築する．CDMは基本的にPMMと類似しているが，照合出力の効率的のために入力番号を遷移ラベルとしてもつ．遷移関数c_gotoは次のように定義される．

c_goto関数：S× {正整数の集合} →S∪{fail}

拡張３の補集合構造を含む規則の処理は，次のPMMの照合アルゴリズム後に説明する．

【PMM の照合アルゴリズム】

入力： 構造列 α =N1N2...Nn; goto関数，output関数.

出力： α に対するCDM.

方法：

（手順１） {初期設定}

CDMの状態c_stateをPMMの状態に写像する関数mapにより，map(C1)=P1なる初期状態C1をCDMに作成し，入力構造の番号iを1にセットする．また，CDMの状態番号を格納するqueueを{C1}にセットする．

（手順２） {PMM上の遷移確認}

queueの全ての状態c_stateに対して，

(3-1) goto(map(c_state), '*', N_i)=nextstate

(3-2) goto(map(c_state), '+', N_i)=nextstate でc_stateに至る遷移ラベルがi-1[注2] である．

を満足する場合それぞれで次を実行する．

c_goto（c_state，i)=c_nextstate

map(c_nextstate)=nextstate

を作成し，queueにc_nextstateを加える．

（手順３） {入力形態素の移動}

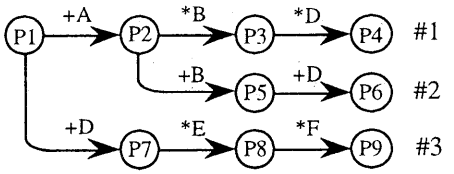
iをインクリメントし，手順２に帰る．但し，最終構造(i=n)になれば，ＣＤＭを出力し，照合を終了する． （アルゴリズム終了 ）

図３にキー集合RULE_SET={A*B*D, A+B+D, D*E*F}に対するPMMのgoto関数を示す．#1,2,3は規則番号である．入力構造

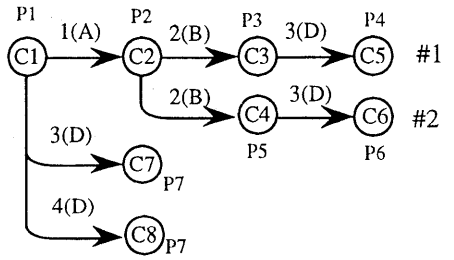
N_1(=A)N_2(=B)N_3(=D)N_4(=D)の照合を考えてみる．CDMの構成を図３-(b)に示す．但し，遷移ラベルは，入力位置（構造）で示し，状態番号の近くにmap関数で写像されるPMMの状態番号を示す．

まず，手順１より，CDMには初期状態C1が作成され，queue={C1}が初期設定される．手順２で入力構造N_1(=A)に対して，PMMは

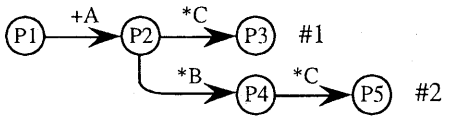
map(C1)=P1； goto(P1, '+', N_1)=P2



(a) PMMの例



(b) CDMの例



(c) 補集合規則の分割例

図３ 照合マシンの例

を定義するので，

c_goto(C1, 1)=C2； map(C2)=P2

が構成され，queue={C1,C2}が得られる．

次の入力構造N_2(=B)に対して，PMMは

map(C2)=P2

goto(P2, '*', N_2)=P3

goto(P2, '+', N_2)=P5

を定義し，C2に至る遷移ラベルは現在位置2の直前1であるので，両方の遷移がCDMに構成される．

c_goto(C2, 2)=C3; map(C3)=P3

c_goto(C2, 2)=C4; map(C4)=P5

queue={C1,C2,C3,C4}に対して，同様の処理を繰り返して図３-(b)のCMDが得られ，

N_1(=A)N_2(=B)N_3(=D)

が規則１と２(#1,2)と照合し，N_3(=D)とN_4(=D)が規則３の一部と照合する．

補集合規則A*!B*Cの!Bは，Bの情報が欠落しているパターンを検出するが，この拡張３を実現するために，補集合規則は図３-(c)の用に分割展開して，PMMに定義される．この機械において，#1は，#2が検出されてないときに，照合されたものとして処理できるので，補集合規則が検出できる．

[注2] 初期状態C1に対しては，このi-1の条件は不要である．

4．各種応用分野に対する動作の定義

規則で定義されたパターンを検出した後，構造列に対して種々の処理を実行する．従って，Rule(p)で定義された構造列

M_{p1}，M_{p2}，…，M_{pnp}（$1 \leqq i \leqq np$）

と対応する照合検出された構造列

N_{p1}，N_{p2}，…，N_{pnp}

がそれぞれ，引数として参照できる動作(action)を定義することが重要である．

例として，文章校正で有用な分離型の重ね言葉の検出と訂正案の表示を考えてみる．記述を簡単にするために，形態素の後の（）内に形態素記号を記入する．

例えば，

"本(N1)を(N2)静かに読書する(N3)"

は，下線部の語彙が重なると不自然であるので，"本を静かに読む"に訂正した文を利用者に提示したい．従って，次の置換処理関数

Replace(f(N3,STR),"読む",f(N3,TENSE))

を動作として導入する．ここで，第3引数は，"読む"の活用とテンスアスペクトへの制約を意味する．また，翻訳処理の効率化に対して

"彼と連絡(N1)を(N2)取る(N3)"

なる文では，機能動詞"取る"を除去して文"彼と連絡する"とするために，次の削除関数

Delete(f(N3,STR))

を導入する．同様に，"連絡"を"取る"のテンス・アスペクトを融合させて，置換関数が次で定義できる．

Replace(f(N1,CAT),"動詞",f(N3,TENSE))

形態素処理を考えると，以上の動作で検出パターンの基本的処理は実行できるが，引数としては，規則定義と同様に辞書アクセス関数も導入すべきである．

5．実験結果

以上の，文章校正，キーワード抽出，文短縮に利用できる定型表現を分析収集し，それぞれの規則を構築した．構築された規則の数は，それぞれ９６，１９３，６０４となった．この実験で，最も多い規則をもつ文短縮に対する処理では，ＥＤＲの日本語コーパス75,000文（約５MB）に対して，実験を行った(8)，(9)．規則データの容量は，約90KBであり，照合回数は29,319回であった．図４に１規則あたりの平均探索時間を示すが，非常に高速であることが分かる．

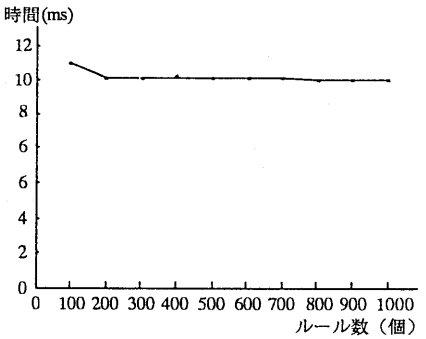


図４　１規則あたりの平均探索時間

6．　むすび

以上，日本語定型表現を応用分野の利用形態により分類・分析し，定型表現の規則を定義した．また，その規則集合に対する照合アルゴリズムを提案し，実験結果で提案手法の有用性を評価した．

今後は，構文情報の木構造の定型表現，格構造の定型表現など各解析レベルで抽出できる情報で利用できる定型表現の分析を行い，本手法の応用を検討したい．

【参　考　文　献】

(1) Aho, A.V., and Corasick, M.J., Efficient string matching: An aid to bibliographic search, *C.ACM*, 18, 6, pp.333-340 (1975).

(2) Aoe, J., An efficient digital search algorithm by using a double-array structure, *IEEE Trans. Softw. Engr.*, SE-15, 9, pp.1066-1077 (1989).

(3) Aoe, J., Morimoto, K., and Sato, T., An efficient implementation of trie structures, *Softw. Prac. & Exper.*, 22, 9, pp.695-721 (1992).

(4) 島津，内藤，野村："助詞「の」が結ぶ名詞の意味関係のsubcategorization" 情処，NL研報,53-1 (1986).

(5) 白井，池原，河岡，中村："日英機械翻訳における原文自動書き替え型翻訳方式とその効果"情処論，36，1，pp.12-21(1995).

(6) 新納，井佐原："疑似Nグラムを用いた助詞的定型表現の自動抽出" 情処論，Vol.36(1995-1).

(7) 首藤："文節構造モデルによる日本語の機械処理に関する研究"福岡大学研報45自然科学編,6,pp.88-119 (1980-3).

(8) 津田，中村，青江："形態素置換による文書短縮信学論 (D-Ⅱ), J75-D-Ⅱ.No3.pp.619-627 (1992-3).

(9) 辻，安藤，獅々堀："活用語を含む助詞的定型表現の分析" 情処５２回全国大会 (1996-3).

(10) 林，菊井："日本文推敲支援システムにおける書き換え支援機能の実現方式" 情処学 Vol.32 (1991-8).

(11) 宮崎："係り受け解析を用いた複合語の自動分割法" 情処学 Vol.25 (1984-11).