日本語処理基本システム（２）

―辞書検索系と構文解析系―

2C-2

菅野　祐司　　長尾　健司　　上田　謙一
松下電器産業株式会社　東京研究所

１．はじめに

本稿では、我々が現在構築中の日本語処理基本システム[1]のうち、辞書検索系と構文解析系について、その特徴、処理方式、性能評価、問題点等を報告する。

２．辞書検索系の特徴と処理方式

本システムが用いる３種類の辞書のうち、自立語辞書はその規模を考慮し、２次記憶上に格納する。この制約の下で、日本語の特徴の一つである、大きな表記の自由度に対処できる自立語辞書の検索系を考案した。

この種の辞書検索系として京都大学の方式[2]が知られているが、索引の容量が７万語の辞書に対し７００ＫＢと大きいこと、混ぜ書き表記に対しては検索処理に時間がかかること等の問題があった。本方式は上記の方式をもとに性能の向上を図ったもので、次の特徴を有する。

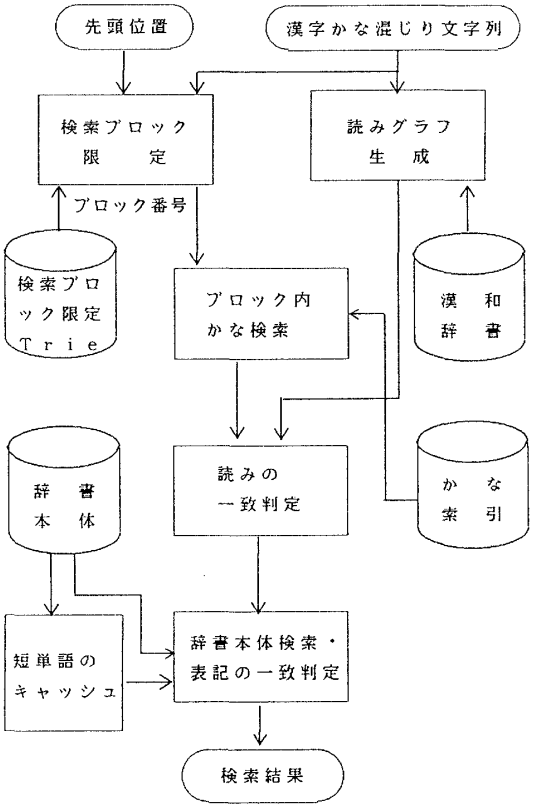

図１．辞書検索系の全体構成

Ａ）かな，混ぜ書き，正書の全ての表記での効率的検索
Ｂ）形態素解析の際の検索のされ方を考慮した効率化
Ｃ）辞書本体や索引情報の容量のコンパクト化

図１に示す構成図にそって、辞書検索系の処理方式を説明する。まず、検索の前に、辞書の原データから次の４種のファイルを作成しておく。

Ａ）辞書本体
　辞書本体は単語の読みの順序に配列した固定長ブロック（ブロック＝２次記憶へのアクセス単位）のファイルで、ブロック長はレコード長（レコード＝各単語の語彙情報）の数十倍とする。各ブロック内のレコード数は可変とし、ブロックの切れ目を特定するかな文字列の長さが短くなるように調整する。
Ｂ）かな索引
　ブロックの切れ目を特定するかな文字列と、全ての異なるかな見出しの索引を高圧縮した形で作成する。
Ｃ）漢和辞書
　見出し語に現れる全ての漢字（列）と、その読みを記述する。
Ｄ）検索ブロック限定Ｔｒｉｅ
　入力漢字かな混じり列の先頭部から検索すべきブロックを小数個に限定するためのTrieで、図２にその一例を示す。図２は、例えば入力の先頭２文字が、「故郷」の場合には３７０，３８５，１０２９の３つのブロック内を検索すればよいことを表す。

```
惑 (826)
故 (1202)
  さ
    と (1029)
  郷 (385,1029)
  山 (391)
  買 (400)
枯
  か (384)
```

図2．Ｔｒｉｅの例（一部）

上のＢ）からＤ）及びキャッシュのロードの後、入力文を漢字かな混じり表記の文字列で登録する。その後は、検索する部分列の先頭位置を与えるたびに、その最左部分列と表記、読みが適合する全ての語の辞書情報が出力される。入力文の登録時には、漢和辞書を引いて入力文の全ての可能な読み方を求めておく。先頭位置が入力されると、部分列を鍵にTrieを検索し、検索すべきブロック番号（一般に複数）を求め、その各ブロック内の語をかな索引から順に検索し、読みグラフと照合する。
一致した場合には辞書本体を検索して表記まで含めた照合を行い、一致したものが検索結果として出力される。このとき、単語がキャッシュにあれば辞書本体のアクセスは行わない。

３．辞書検索系の性能評価と問題点

以上の辞書検索系を九州大学基本語辞書[3]（見出し語数は約8万語）に対して適用し、性能評価を行った。九大

Core System for Japanese Processing (2)
- Dictionary Retriever and Parser -
Yuji Kanno, Kenji Nagao, Kenichi Ueda
Tokyo Research Laboratory
Matsushita Electric Industrial Co.,Ltd.

辞書には正書表記とかな表記の各文字の間の「対応」情報があり、検索ブロック限定Trie及び漢和辞書はこれを用いて自動作成した。　辞書のブロックへの分割結果は、1ブロック中の最大格納語数が70語の場合に、1ブロック当りの平均語数が53語（充填率76%），ブロック間の区分文字列の平均長が2.4文字，最大長が4文字と、充填率は多少低いが区分文字列は十分短い。　Trieについては、ノード数は約2万，総容量は135KB，平均深さは2.02であり、コンパクトで浅いTrieとなった。ただし、最大深さは5で、「一」のような読みの多い漢字で始まる語に関しては詳細な情報を盛り込まれている。また、かな索引は「かな表記，同音異議語の数，辞書本体の位置」の3種の情報を複合Huffman符号を用いて1語18bitに圧縮した。かな索引の総容量は140KBであった。検索速度は、VAX8650上で平均50文字／秒で、混ぜ書き表記の場合でも、正書表記の場合の1/2以上の速度で検索が可能であった。

検索時間短縮をはかるためには、読みグラフ作成とかな索引の検索の高速化及び辞書本体の主記憶上への展開を図る必要がある。また、表記のゆれの一種である、送り仮名のゆれの吸収も今後の課題である。

4．構文解析系の特徴と処理方式

本システムの形態素解析結果である、複数の解釈を含む有効グラフをそのままの形で効率よく処理できる構文解析系を考案した。以下にその特徴を列挙する。

A）入力は意味構造の付与された品詞をノードとし、接続関係をアークとするグラフで、文法規則は意味構造を操作するための拡張部が付加された分脈自由文法の形式とする。出力は解析表であるが、文カテゴリに付随する意味構造のみで十分な場合も多い。
B）複数の意味を持つ単語，句の意味構造を表現可能な意味表現形式[1]を採用する。
C）品詞，終了位置が同じで、他の（意味的な）情報が異なる複数の形態素（句）群を１つの実体（解析表の要素）として扱い、形態素間の差異は意味構造で表現する。
D）文法規則の拡張部では、左辺カテゴリの意味構造を受け取り、右辺カテゴリの意味構造を生成するが、上記の意味構造を操作するための専用関数群を用いて複数の意味（解釈）の同時並列的な操作を行う。

構文解析の基本アルゴリズムはEarleyの方法[4]をもとに、意味構造及びグラフ構造を扱うための次のような拡張を施したものである。

A）入力形態素グラフから、品詞名を名前とする意味フレーム形式の形態素の集合を作る。このとき、開始，終了位置は、入力文の文字の位置ではなく、図３の様に、グラフ上の接続関係を表現する仮想的な位置とする。その後、品詞と終了位置を共有する形態素をマージして１個の形態素のように扱う。結果的に、開始位置は複数個の並びになる。
B）解析表の要素は、
（＜右辺の未処理カテゴリ数＞
＜右辺カテゴリの並び＞
＜開始位置並び＞　＜終了位置＞　＜意味構造＞）
である。ここで、意味構造は、左辺カテゴリ名を名前とする意味フレーム[1]とする。
C）解析表作成の基本は次の点を除き従来通りとする。
・文末から文頭へ向かって走査し、ある位置の解析表が空でも失敗とせず、処理を続行する。
・「予測付加」の直前に、品詞及び終了位置が同一である解析要素のマージを行う。
・部分予測（L->R1‥Rj*Rj+1‥Rn）の処理では、開始位置が複数個あることが問題となるが、開始位置の最大値が着目位置と等しい場合に処理を行う。
・拡張部では意味操作関数群を用いて、意味フレームの解釈の分枝を限定するが、その結果である開始位置の変化を解析表に反映させる。

5．構文解析系の評価と問題点

現在、構文解析系の評価のため、基本的な日本語の文法規則をインプリメントしてテストを行っている。その一例を図４に示す。現状での最大の問題点は、複数の解釈の同時並列的な処理を行うプログラムが記述しにくい事で、日本語の文法記述に適した意味操作関数群を文法作成と同時に、半ば試行錯誤的に構築し、文法記述のスタイルを確立して行きたい。

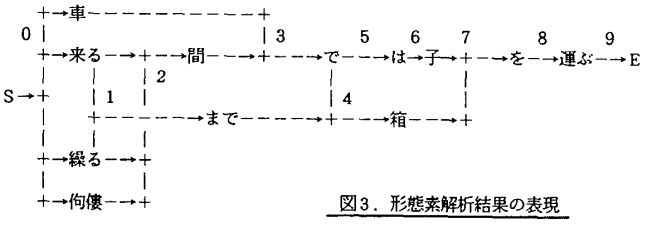

図3．形態素解析結果の表現

図4．文法規則の一部

```
((単位文 -> 格要素 単位文)
 (lambda (r1 r2)
           .
           .
           .
 (破壊的系列値組変更 格要素意味 '(格要素 体連語) '(格要素 格表示 格パターン)
   '(lambda (v1 v2)
      (let ((%動詞格情報 nil))
        (破壊的系列値組変更 '(格パターン 格表記) '(格パターン パターン)
          '(lambda (v3 v4)
  ;
  ;          ひとつの格要素の格パターンのうち
  ;        動詞との整合性を考慮して可能なもの保持しておく。
  ;
              (破壊的系列値変更
                   動詞命題文核部 (並び生成 '動詞命題文核部 'ヴォイス)
                   '(lambda (v)
                          :
                          ;          格の整合性チェック及び
                          ;        体連語埋め込みは，整合性
                          ;        が否定的な場合 nil を返す
                          ;
```

6．おわりに

我々が構築中の日本語処理基本システムのうち、辞書検索系と構文解析系について、特徴，処理方式，評価及び問題点を述べた。今後はシステムの機能，性能の向上を図ると共に、実証規模の辞書，文法，用例データを用いて評価を進めてゆきたい。最後に、本システムの構築に当たって、幅広く御指導を頂いている筑波大学の荻野綱男講師に深く感謝します。

【参考文献】

1）長尾:"日本語処理基本システム（１）",本大会
2）長尾,辻井他:"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,(1978)
3）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,(1982)
4）長尾:"日本語情報処理",電子情報通信学会刊,(1984)