軽量・高速な日本語解析ツール『簡易日本語解析系Q_JP』

亀田 雅之
(株) リコー・研究開発本部・情報通信研究所
kameda@ic.rdc.ricoh.co.jp

概　要

従来の大規模な日本語解析系に対し、多くの応用系に手軽に利用できる実用レベルの軽量・高速な日本語解析ツール『簡易日本語解析系 Q_JP』を開発した。

Q_JP は、機能語と字種に着目した小規模辞書ベースの形態素解析系と、品詞情報ベースのヒューリスティック規則により係り受け選択を行う構文解析系からなる。1. **軽量性**［辞書：100KB 足らず，メモリ：約250KB(DOS 版)］、2. **高速性**［PC(80486/25MHz)：80〜150 語/秒，WS(Sun-SS20)：700〜800 語/秒］、3. **頑強性**［長文解析，未知語登録不要］を備え、パソコンでも余裕をもって動作する。

Q_JP の機能はライブラリ化し、現在、文書検索、読解支援、自然言語 I/F 等への応用を図っている。

1　はじめに

日本語解析技術は、今後の高度な日本語文書処理のキーとなる重要な技術である。

しかし、日本語解析のためには、数万語以上の大規模な辞書や曖昧さや意味を扱う重い処理系が必要である。また、未知語やその登録作業、曖昧さの組合せ爆発のために長い文が扱えない、といった問題がある。

こうした問題を踏まえ、多くの応用系に手軽に利用できる実用レベルの日本語解析系を目指し、AWK言語上で試作した『簡易日本語解析系 Q_JP』を報告した [1]。試作システムでは、小規模辞書で形態素解析から構文解析 (係り受け解析) まで行ない、かつ、十分な解析性能と頑強性をもつことを示した。その後、さらにC言語で実装することにより、少ないメモリで極めて高速に解析する系を実現し、所期の目的であった応用系への実用レベルでの展開を可能とした。

Q_JP では、日本語文における字種の特徴に着目し、未知語の問題を避けながら、機能語主体に辞書を小規模化するとともに、構文解析での曖昧さの扱いを簡易化し、意味に踏み込まない品詞情報レベルで優先選択することで、処理を軽くしている。

日本語解析系としては、形態素解析系 JUMAN[2] や構文解析系KNパーサ [3] 等が公開されており、また、各所で汎用化を図っていると思われるが、Q_JP のように応用系への組み込みを指向し、軽量性、高速性、頑強性を備えた日本語解析系の提示はない。

本稿では、Q_JP の解析方式を概観し、現状のC言語版の諸元について示す。

2　Q_JP の解析方式

Q_JP は、漢字かな混じり日本語文を形態素列, 単語列に分割する形態素解析系、さらに係り受け構造に解析する構文解析系からなり、各々次の方針を採った。

- 字種の特徴を利用した文節分割 [4] や予備的な単語分割 [5] の方式を、小規模辞書ベースの形態素解析系として拡張・精緻化する
- 構造の曖昧さを簡易的に扱うとともに、小規模辞書を生かすため、意味情報に依らず品詞/構文レベルの情報に基づく構文解析系とする

2.1　形態素解析系

形態素解析系では、品詞ごとの 表記の字種 (漢字/ひらがな/カタカナ等)、語彙の数、接続上の制約 の各特徴、及び和語系複合語等の形態素構成の特徴に着目し、形態素候補を次の 辞書、字種-品詞割り当て、単語合成 の分担で扱い、辞書を小規模化する。

辞書：辞書には、語彙的に閉じた系をなす機能語 (助詞, 助動詞, 接続詞, 連体詞, 副詞等) や活用語尾等を中心に、少数品詞語、次の字種-品詞割り当てから外れる例外語等を登録する。

字種-品詞割り当て：辞書引きできなかった同一文字種の文字列に対し、字種と文字列長に応じて、語彙数の多い概念語品詞 (名詞, サ変名詞や動詞, 形容動詞の語幹) を候補として割り当てる。語彙数の多いこれらの品詞の単語を字種-品詞割り当てで扱うことで、登録単語数を大幅に削減する。

単語合成: 日本語では、和語系の複合語 (「組み-込む」, 「切れ-味」)、派生語 (「書か-す」)、品詞転性 (「歩み」) 等の造語力が顕著であり、辞書エントリを増やす要因でもある。これらに対して、短単位の形態素解析と、活用語の語幹と語尾の合成を行う単語合成の枠組みの中で、複合語の形態素の並びを検出し、単語として認識することで、辞書への登録を不要とする。

図１に、短単位の形態素解析結果と単語合成の例を示す。図中の「日本語」他の漢字表記の名詞と「切(る)」/「置(く)」の漢字の動詞語幹、和語複合語「切れ目」は辞書にない。

上記の字種-品詞割り当ては仮想辞書引き処理であり、形態素解析手法は、最長一致と品詞接続検査による方法と変わらない。尚、接続検査で品詞の曖昧さが未解消の形態素に対しては、前後の品詞や単語を検査して、曖昧さを解消したり、優先順位を変えるヒューリスティックな品詞の曖昧さ解消規則がある。

辞書は４種類ある (表１) が、通常は、拡張辞書 (非ひらがなの機能語, 例外語等) とひらがな辞書 (ひらがな機能語, 活用語尾, 主要なひらがな単語) を参照する。ひらがな拡張辞書は、ひらがな文字列部分の解析失敗でのバックトラック時に、形容動詞語幹辞書は、形容動詞を含む品詞の曖昧さ解消時に参照される。

【形態素リスト】

```
*[ 1]( 8)日本語     [zJ] (523)名詞
.[ 2](14)の         [zH] (37)ノ-格助
.[ 3](16)よう       [zH] (293)ヨウダ-ダ幹
|[ 4](20)に         [zH] (250)ダ用b

*[ 5](22)単語       [zJ] (523)名詞
.[ 6](26)間         [ X] (356)形式名詞
.[ 7](28)に         [zH] (36)ニ-格助

*[ 8](30)切         [zJ] (349)下ラ基
|[ 9](32)れ         [zH] (351)下ラ用
+[10](34)目         [ X] (507)単独名詞
.[11](36)を         [zH] (41)ヲ-格助

*[12](38)置         [zJ] (369)五カ幹
|[13](40)か         [zH] (372)五カ未a
.[14](42)な         [zH] (256)ナイ-ク幹
|[15](44)い         [zH] (205)ク体

*[16](46)膠着言語   [zJ] (523)名詞
.[17](54)の         [zH] (37)ノ-格助

*[18](56)文         [zJ] (523)名詞
.[19](58)の         [zH] (37)ノ-格助

*[20](60)処理       [zJ] (523)名詞
.[21](64)に         [zH] (36)ニ-格助
.[22](66)お         [zH] (184)オク-五カ幹
|[23](68)い         [zH] (375)五カ用b
.[24](70)て         [zH] (72)テ-接助
.[25](72)、         [zT] (512)読点

*[26](74)形態素解析 [zJ] (523)名詞
.[27](84)は         [zH] (47)ハ-係助

*[28](86)第         [ X] (502)数名詞-頭
|[29](88)一         [zN] (501)数名詞
.[30](90)の         [zH] (37)ノ-格助

*[31](92)関門       [zJ] (523)名詞
.[32](96)で         [zH] (243)ダ-用c
.[33](98)あ         [zH] (170)アル-五ラ幹
|[34](100)る        [zH] (434)五ラ終
.[35](102)。        [zP] (353)句点
```

【単語リスト】

```
*[ 1]( 8)日本語     (日本語)    (41)名詞
.[ 2](14)の         (の)        (51)ノ-格助
.[ 3](16)よう-に    <ようだ>    (60.22)ヨウダ-ダ用b

*[ 4](22)単語       (単語)      (41)名詞
.[ 5](26)間         (間)        (43)形式名詞
.[ 6](28)に         (に)        (51)ニ-格助

*[ 7](30)切-れ-目   (切れ目)    (41)名詞
.[ 8](36)を         (を)        (51)ヲ-格助

*[ 9](38)置-か      <置く>      (15.11)動:五カ未a
.[10](42)な-い       <ない>      (60.4)ナイ-ク体

*[11](46)膠着言語   (膠着言語)  (41)名詞
.[12](54)の         (の)        (51)ノ-格助

*[13](56)文         (文)        (41)名詞
.[14](58)の         (の)        (51)ノ-格助

*[15](60)処理       (処理)      (41)名詞
.[16](64)に         (に)        (51)ニ-格助
.[17](66)お-い      <おく>      (75.22)オク-五カ用b
.[18](70)て         (て)        (55)テ-接助
.[19](72)、         (、)        (92)読点

*[20](74)形態素解析 (形態素解析) (41)名詞
.[21](84)は         (は)        (52)ハ-係助

*[22](86)第一       (第一)      (46)数名詞
.[23](90)の         (の)        (51)ノ-格助

*[24](92)関門       (関門)      (41)名詞
.[25](96)で         <だ>        (60.23)ダ-用c
.[26](98)あ-る      <ある>      (75.3)アル-五ラ終
.[27](102)。        (。)        (91)句点
```

図１. 形態素解析結果例
[形態素リスト/単語リスト]

〔例文出典〕情報処理学会 第 42 回全国大会 予稿集, 1991

2.2 構文解析系

構文解析では、文節間の係り受け解析をベースとし、係り受け文節対の組合せの枠組み [6] で、曖昧さを簡易的に扱う。これは、構造的な全解生成 (積算的組合せ) を文節ごとの係り先の可能性保持 (加算的組合せ; 図５) で代替するもので、組合せ爆発によるメモリ不足/処理過多を回避し、長文も解析可能とする。

日本語構文解析では、一般に意味情報による優先選択が用いられるが、効果は必ずしも大きくない。また、Q_JP では、意味情報を持つことは、形態素解析系の小規模辞書の特徴と反することになる。そこで、係り先優先選択では、意味情報を用いず、係り受け可能な最近接文節を優先しながら、読点や副詞性単語の有無、表層的/構造的な類似性 [7] といった品詞/構文レベルの情報 [8] を基にしたヒューリスティック規則で、必要に応じて係り先を付け変えるという方式をとる。

構文解析系の全体は、文節単位に、品詞構成から文節属性を設定し、文節属性の組合せから係り受け可能文節対を検出し、さらに係り先文節を選択するという流れである。(図３～５)　各処理は、対応する規則に基づいて行われる。尚、係り先が認定できなかった場合に、形態素解析まで戻って、品詞の再選択を行い、構文解析をやり直す係り受け失敗回復機構をもつ。

3 Q_JP ライブラリ

3.1 ライブラリ構成

Q_JP は、C言語で実装し、応用系に容易に組み込めるようライブラリ化した (DOS 版,UNIX 版)。ライブラリは、形態素解析系 qjp_s 関数、構文解析系 qjp_k 関数を始めとする主要関数群と各種解析結果の情報等を参照/表示する補助関数やマクロ群からなる。

表１に、現版の辞書等のサイズを示す。辞書の合計は、約 4500 語/50KB 足らずである (今後、例外語 [拡張辞書] や和語のひらがな表記語 [ひらがな拡張辞書] 等を追加する必要があるが、その場合でも全体で 100KB

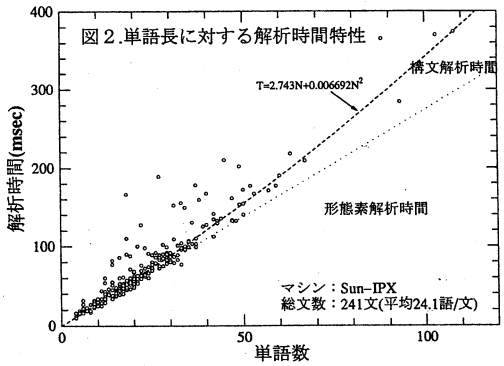
表１. Q_J P (DOS 版) のファイルサイズ

ワークベンチ実行ファイル	151.2KB	合計 228.4KB
制御テーブルファイル	31.8KB	
実行時辞書ファイル (小計)	45.4KB	4476語/3839レコード
拡張辞書	18.7KB	1836語/1469レコード
ひらがな辞書	5.6KB	740語/ 374レコード
ひらがな拡張辞書	10.9KB	869語/ 865レコード
形容動詞語幹辞書	10.2KB	1131語/1131レコード

程度であると見込んでいる)。辞書が小さいため、辞書データをメモリ上に展開する高速モードが可能となる。約30KBの制御テーブルには、形態素解析の533形態素品詞間の接続表、字種-品詞割り当て規則、接辞テーブル、辞書インデクス等が格納されている。

この他、形態素解析系は、文節内接続型規則・品詞曖昧さ解消規則・単語合成規則、構文解析系は、文節属性設定規則・係り受け可能文節対検出規則・係り先優先選択規則・係り受け失敗回復規則をもつ。これらは、IF-THEN形式のC言語関数として実装している。

上記の解析系系に対話/表示系を加え、対話型/バッチ型で利用できる日本語分析ツール「Q_JPワークベンチ」を作成した。ワークベンチの実行モジュールサイズは約150KBであり(DOS版)、辞書、制御テーブルを含めた全体でも230KBと小さい(表1)。尚、対話系には、ユーザが品詞の曖昧さ解消、係り先の変更を指示する機構を備える。



3.2 解析実験

「Q_JPワークベンチ」により、試作システムと同様に2コーパス[1:平均24.1語,241文/2:平均29.5語,210文]で解析実験を行った[1]。

実行性能

実行時メモリは、DOS版で約245KB(高速モード:+25KB)、UNIX版で約500KBであった。このメモリで約60文節までの長文の解析を行うことができる。

解析速度は、パソコン(80486/25MHz)で80～150*語/秒、ワークステーション(Sun-SS20)で700～800*語/秒であり(*:高速モード)、50文節文(図3～5)をパソコン(80486)で1秒未満で解析できる(入出力時間を含まず)。図2に単語長に対する解析時間の特性を示す。構文解析時間は、文長の二乗オーダとなるが、係数が小さいため、100単語程度の文長の範囲では形態素解析の線型の時間が大部分を占め、実用の範囲では解析処理全体としてもほぼ線型の特性になる。

比較できる構文解析系の実行性能データはないが、形態素解析系JUMAN 2.0の1MBテキストの解析時間76～93分, メモリ5MB[2](マシン不明)と比較すると、Q_JPは、構文解析まで行なって、1MBテキスト換算で解析時間約6分(9秒/約200文25KB), メモリ0.5MB(SS20)であり、各々1桁小さい。

解析性能

解析正解率は、若干改善しているが、試作システムの正解率[1][コーパス1/2に対し、形態素正解率:99%/95%, 文節係り先正解率:95%/90%, 1文全体の係り受け正解率:71%/41%]とほぼ同程度である。

【係り受け木構造】

```
 1[50](13) :機械翻訳方式
 2[49](12) : └する
 3[48](11) :   ├特徴と
 4[47](11) :   └ことを
 5[46](10)>:     └生成する
 6[45]( 9)>:       ├[他言語の]文を
 7[43]( 9)?:       ├[その]概念構造から
 8[41]( 9)>:       └定め、
 9[40]( 8)>:         ├[[[[翻訳すべき]言語から]なる]文の]概念構造を
10[35]( 8)?:         ├[[[既に]選択された|概念記号と]意味関係記号に]基づき、
11[30]( 8)?:         └設け、
12[29]( 7)>:           ├[[[[利用者が]選択する|概念記号及び]意味関係記号を]受け入れる]手段を
13[23]( 7)?:           └翻訳システムにおいて、
14[22]( 6)>:             └翻訳する
15[21]( 5)>:               ├目標言語に
16[20]( 5)?:               ├[[[[その]概念構造に]基づいて]翻訳すべき]言語を
17[15]( 5)?:               └定め、
18[14]( 4)>:                 ├概念構造を
19[13]( 4)?:                 ├[[[各単語に]対応する]概念記号と、|[[[各概念記号間の]関係を]示す]意味関係記号から
20[ 6]( 4)?:                 └分割し、
21[ 5]( 3)>:                   ├単語に
22[ 4]( 3)?:                   └[[[翻訳すべき]言語から]なる]文を
```

【原文】

翻訳すべき言語からなる文を単語に分割し、各単語に対応する概念記号と、各概念記号間の関係を示す意味関係記号から概念構造を定め、その概念構造に基づいて翻訳すべき言語を目標言語に翻訳する翻訳システムにおいて、利用者が選択する概念記号及び意味関係記号を受け入れる手段を設け、既に選択された概念記号と意味関係記号に基づき、翻訳すべき言語からなる文の概念構造を定め、その概念構造から他言語の文を生成することを特徴とする機械翻訳方式

図3. 50文節文の係り受け縮退木構造

〔例文出典〕公開特許公報(昭60), 1985

4 まとめ

字種の特徴を利用した小規模辞書による形態素解析と、品詞レベルの構文情報による係り受け解析ベースの構文解析に基づく簡易日本語解析系 Q_JP をC言語で実装した。試作版で、小規模辞書下で高い解析性能を示したが、さらに、メモリ、速度も実用レベルになった。従来の解析系に比べて、はるかに軽量・高速で、未知語や長文に対する頑強性も備えている。

Q_JP はライブラリ化しており、多くの応用系に手軽に組み込むことができる。現在、文書検索、読解支援、自然言語 I / F等への応用を図っている。また、構文レベルでのコーパスの分析にも利用している。

尚、現在、Q_JP では、漢字連続の名詞複合語は分割せず、応用系側に託しているが、将来は、接辞や文字連接の統計情報に基づく [5, 9] 等、辞書に依らない分割機能を付加することを検討している。

図4．50 文節文の
文節詳細情報

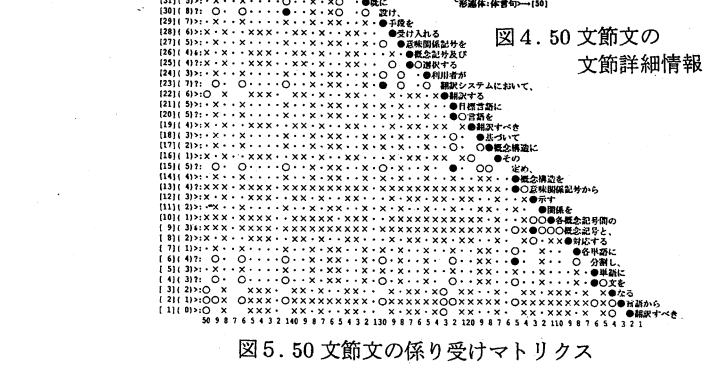


図5．50 文節文の係り受けマトリクス

参考文献

[1] 亀田雅之：簡易日本語解析系 Q_JP, 情報処理学会 自然言語処理研究会報告 94-4, 1993.

[2] S.Kurohashi, T.Nakamura, Y.Matsumoto and M.Nagao : Improvement of Japanese Morphological Analyzer JUMAN, Proc. of the International Workshop on Sharable Natural Language Resources, 1994.

[3] 黒橋禎夫

[4] 坂本義行： 文節単位の自動分割法, 計量国語学 11-6, 1978.

[5] 長尾真, 辻井潤一, 山上明, 建部周二：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, pp.514-521, 1978.

[6] 亀田雅之, 石井信, 伊東秀夫： 曖昧性解消のための対話機構を備えた日本語解析系, 情報処理学会 自然言語処理研究会報告 84-2, 1991.

[7] 黒橋禎夫, 長尾真： 長い日本語文における並列構造の推定, 情報処理学会論文誌 Vol.33 No.8, pp.1022-1031, 1992.

[8] 兵藤安昭, 池田尚志： 表層的情報による日本語長文の骨格構造解析, 情報処理学会 自然言語処理研究会報告 103-15, 1994.

[9] 武田浩一, 藤崎哲之助：統計的手法による漢字複合語の自動分割, 情報処理学会論文誌 Vol.28 No.9, pp.952-961, 1987.

補足

最近のハードウェアやOSの進歩により、メモリやディスク容量、速度の制約が急速に小さくなってきており、Q_JP の小規模辞書を始めとする軽量・高速性に疑問が呈されることがある。これに対する筆者の指針を示しておきたい。

- 解析系の実現だけが目的なら、必ずしも軽量である必要はないが、Q_JP は応用系に組み込まれる一部分であり、本来の目的は応用系にある。制約緩和のメリットは、応用系全体で享受するもので、部分系たる Q_JP が多くを享受するべきではない

- Q_JP を解析プリプロセサと位置付け、応用系側では対象分野の十分な辞書を用意し、解析のレベルを深めるという分担が有効である（係り先の可能性は残してある (図5) ので、応用系で意味情報等を使って係り先を変更する等もできる)。また、解析系の辞書と応用系の品詞体系との整合等を取るのが難しかったり、解析系の辞書に応用系の情報を設定する枠組みがないと、解析系と応用系が各々大規模な辞書を抱えなくてはならない。こうした場合、解析系側の辞書が小さく負担がないことは意義がある

いずれにせよ、Q_JP により、応用系は、従来に比べて極めて小さな負担で構文解析レベルの情報を手にすることができる。この点で、日本語解析の応用展開を促進する効果は十分にあると考える。