# A Portable & Quick Japanese Parser : QJP

Masayuki KAMEDA
Information and Communication R & D Center, RICOH COMPANY, LTD.
2-3, Shin-Yokohama 3-chome, Kohoku-ku, Yokohama, 222, Japan
kameda@ic.rdc.ricoh.co.jp

## Abstract

QJP is a portable and quick software module for Japanese processing. QJP analyzes a Japanese sentence into segmented morphemes/words with tags and a syntactic *bunsetsu kakari-uke* structure based on the two strategies, a) Morphological analysis based on character-types and functional-words and b) Syntactic analysis by simple treatment of structural ambiguities and ignoring semantic information. QJP is small, fast and robust, because 1) dictionary size (less than 100KB) and required memory size(260KB) are very small, 2) analysis speed is fast (more than 100 words/sec on 80486-PC), and 3) even a 100-word long sentence containing unknown words is easily processed.

Using QJP and its analysis results as a base and adding other functions for processing Japanese documents, a variety of applications can be developed on UNIX workstations or even on PCs.

## 1 Introduction

Natural language parser/analyser is essential for allowing advanced functions in document processing systems, such as keyword extraction to characterize a text, key-sentence extraction to abstract a document, grammatical style checker, information or knowledge retrieval, natural language understanding, natural language interface and so on. But a general purpose parser requires 1) a large dictionary database with more than several tens of thousands words, 2) advanced techniques for disambiguation and processing semantics, and 3) substantial machine resources, such as a lot of memory and high speed CPU.

In addition, users must maintain additional terms in dictionaries for specialized fields. As a result, most parsers cannot be easily used in applications and it is difficult to develop a practical parser which can be easily integrated into many applications.

We changed our viewpoint in order to design and develop an applicable and usable Japanese parser. First, we focused on the unique sets of character-types in written Japanese and constructed a very small dictionary using mainly functional words in *hiragana*-character. Similar approaches[1][2] were used for segmentation or preliminary morphological analysis about 20 years ago, using the transition-point between types of character sets to cue word segmentation. Second, we noticed that dealing with syntactic ambiguities creates a large processing burden and even using semantic information does little to assist syntactic analysis at the current level. So we either simplified dealing of structural ambiguities or ignored semantics to lighten the syntactic processing.

We first created a prototype of our parser[3] using AWK language, and then rewrote it [4] in C so it could be included in applications. The resulting parser, named QJP, is portable, fast and robust. It is an effective parser for many general purpose applications, despite of a dictionary size of only 5 thousand words. It can analyze a 100-word sentence on a PC in less than one second, while using less than half of a megabyte of memory. In addition, it requires no further dictionary maintenance for new terms .

In this paper we describe the QJP's analysis methods and report on its current performances.

## 2 Analysis Method

QJP performs two types of analysis : 1) morphological analysis to segment a sentence into part-of-speech tagged morphemes and words, 2) syntactic analysis to place words into *bunsetsu*[1]-dependency structure. Analysis strategies are the followings :

- The morphological analysis is achieved by expanding an earlier methods[1][2] for *bunsetsu* or word segmentation using character-types thus allowing the use of a very small dictionary.
- The syntactic analysis uses no semantic information, only part-of-speech and other syntactic information. In addition, rather than creating all possible, or some preferable, parses, we construct the best syntactic structure preserving local ambiguities.

[1] *Bunsetsu*(文節) is a kind of phrasal unit in Japanese, consisting of one content word(内容語/概念語) [such as noun(名詞), verb-noun(サ変名詞), verb(動詞), adjective(形容詞), verb-adjective(形容動詞) and adverb(副詞)] and successive adjunctive words(付属語) [such as auxiliary verbs(助動詞) and post-positional particles(助詞)], and carring one concept.

## 2.1 Morphological Analysis

### Characteristics of written Japanese

A Japanese sentence has no spaces between words[Figure 1]. So it is difficult to segment a sentence into words. However, the fact that at least four distinct sets of characters [for example, *kanji*(”日”,”本”,”語”,etc), *hiragana*(”の”,”よ”,”う”,etc), *katakana*(”ノ”,”ヨ”,”ウ”,etc) and other characters (alphabets, numbers, symbols etc.)] are used to write Japanese can be used for segmenting words. Most words written in *kanji* or *katakana* are content words, such as nouns, verb-noun and stems(語幹) of verbs or adjectives. Most words in *hiragana* are functional words(機能語), such as postpositional particles, auxiliary verbs and inflective suffix(活用語尾) of verbs and others [Table 1]. And the vocabulary of content words is much larger than that of functional words.

Table 1. Classifications of Japanese Part-of-Speech and their word examples

| 分 類 | | 品 詞 | Part-of-Speech | 表 記 例 Examples |
|---|---|---|---|---|
| 自立語 *1 | 概念語 *4 | 名詞 | noun | 橋,夢,書籍,形態素,語彙(語い),コップ |
| | | サ変名詞# | verb-noun (sahen) | 調査-する,国際化-する,コーチ-する |
| | | 動詞# | verb | 書-く,流-れる,愛-する,はか-る(計-る) |
| | | 形容詞# | adjective | 赤-い,美し-い,よ-い,すばらし-い |
| | | 形容動詞# | verb-adjective | 嫌-だ,高級-だ,典型的-だ,リッチ-だ |
| | 機能語 *5 | 形式名詞 | formal noun | 間,上,こと(事),もの |
| | | 副詞名詞 | adverbial noun | 昨日,午後,朝 |
| | | 副詞 | adverb | 結局,万一,既に(すでに),ゆっくり |
| | | 連体詞 | non-conj. adj. | あの,ある(或る),大した(たいした) |
| | | 接続詞 | conjunctive | そして,しかし,だが, |
| 付属語 *2 | | 並列詞 | | 及び(および),または,あるいは |
| | | 助詞 | particle | が,の,は,こそ,など(等),まで(迄) |
| | | 助動詞# | auxiliary verb | られ-る,だ,な-い |
| | | 補助用言# | aux. functional v. | い-る,あ-る,で-きる,にく-い |
| | | 活用語尾 | inflective suffix | し,く,っ,れる,な,い |
| | | 派生語尾 | derivative suffix | す,る,む,がる |
| 接辞*3 | | 接頭辞 | prefix | 第,御(お/ご) |
| | | 接尾辞 | suffix | 回,的,風,化,視,さ,み |

※ *1:independent word *2:adjunctive word *3:affix
*4:content word(conceptual word) *5:functional word
#: inflective -: segment point between stem and inflection part

### Sharing of Morphemes by Dictionary and Rules

Our strategy is that all functional words, which are few in number, are stored in the dictionary and most content words or their stems in *kanji* or *katakana* are to be extracted and given their part-of-speech candidates based on character-types.

Standard morphological analyser uses a dictionary to obtain morpheme or word candidates. But in our approach, morpheme candidates[2] are extracted either from the morpheme dictionary or using allocation rules based on character-type. For example, if the dictionary look-up fails, the allocation rules extract each sequence of character in which all of the characters belong to the same character set. Then, using the allocation rules, part-of-speech candidates are assigned based on the sequence's character set and length. The candidates are disambiguated by checking connection with the the following morphemes based on the connection table between morpheme parts-of-speech. The following morphemes, in most cases, are functional words or inflective suffixes.

The dictionary contains functional words [such as postpositional particles, auxiliary verbs, formal nouns(形式名詞), adverbial nouns(副詞名詞), conjunctions(接続詞), adverbs and so on], inflective suffixes and exceptional content words which cannot be or are not covered by the allocation rules.

Here are some examples of the allocation rules for 1) 1-*kanji* character sequence, 2) 2-*kanji* character sequence and 3) *katakana* character sequence.

1) noun / stem of *5-dan* verb(五段動詞) / stem of *shimo-1-dan* verb(下一段動詞)
2) noun / (stem of *5-dan* verb) / (stem of *shimo-1-dan*) / verb-noun(*sahen-meishi*; サ変名詞) / verb-adjective(形容動詞)
3) noun / verb-noun / verb-adjective

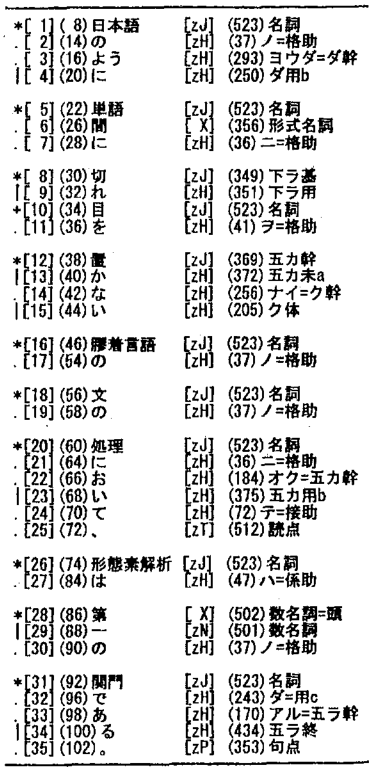
The 1-*kanji* character nouns and verb-stems are largely of old-Japanese-origin words, *wago*(和語), and 2-*kanji* character nouns, verb-nouns and verb-adjectives are mainly Chinese-origin words, *kango*(漢語). In addition, there are several 1-*kanji* character stems of *kami-1-dan* verbs(上一段動詞), *sahen* verbs(サ変動詞) and adjectives(形容詞) which are stored in the dictionary because they are so few in number. The word number of words which can be treated using rules like those given above is so great that the dictionary size is substantially reduced.

### Treating of *Wago* compound words

Another characteristic of old-Japanese-origin verbs (*wago* verb) is that they often continue with other words or morphemes to become verbs or nouns. For examples, two verbs ”書く”(’*to write*’) and ”込む”(’*to become crowded*’) combine into the compound verb ”書き込む”(’*to write into*’), the verb ”読む”(’*to read*’) become the verb ”読ます”(’*cause to read*’) with the causative suffix ”す”, and the verb ”歩む”(’*to step*’) becomes the the noun ”歩み”(’*a step*’) with the derivative suffix ”み”. There are a great many compound words such as these.

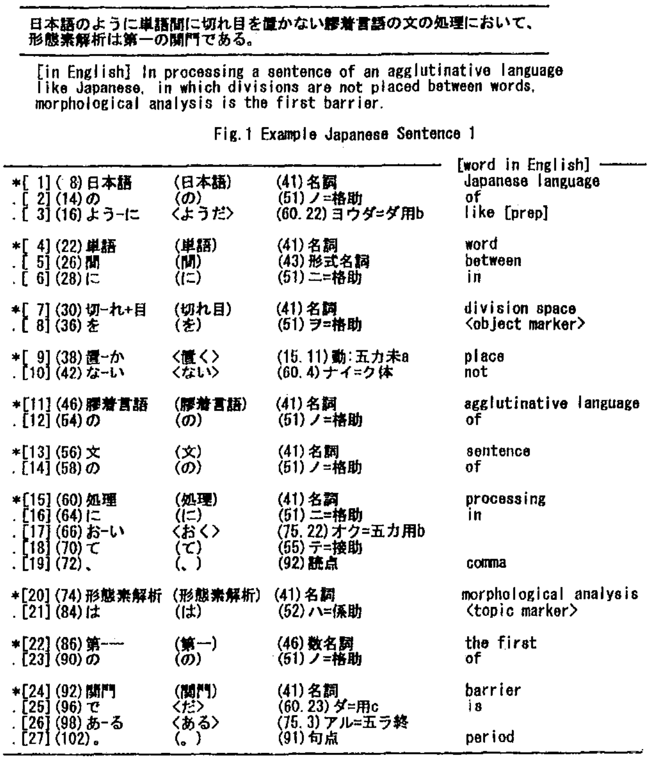
A word-compounding part determines a word from morphemes using word-constituent rules based not only on inflections but also on compounds or derivations such as those shown above. Such rules also greatly reduce the dictionary size.

[2] In this analysis, a inflected word is treated as two or more morphemes - a stem part and one or more inflection part.



```
*[ 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          [zJ]  (523)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
```

Figure 2. Segmented Morphemes with Tags



```
日本語のように単語間に切れ目を置かない膠着言語の文の処理において、
形態素解析は第一の関門である。

[in English] In processing a sentence of an agglutinative language
like Japanese, in which divisions are not placed between words,
morphological analysis is the first barrier.

                    Fig.1 Example Japanese Sentence 1

                                                    [word in English] ——
*[ 1]( 8)日本語     (日本語)     (41)名詞            Japanese language
.[ 2](14)の         (の)         (51)ノ=格助          of
.[ 3](16)よう-に    <ようだ>     (60.22)ヨウダ=ダ用b  like [prep]

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

*[ 7](30)切-れ+目   (切れ目)     (41)名詞            division space
.[ 8](36)を         (を)         (51)ヲ=格助          <object marker>

*[ 9](38)置-か      <置く>       (15.11)動:五カ未a    place
.[10](42)な-い      <ない>       (60.4)ナイ=ク体      not

*[11](46)膠着言語   (膠着言語)   (41)名詞            agglutinative language
.[12](54)の         (の)         (51)ノ=格助          of

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

*[15](60)処理       (処理)       (41)名詞            processing
.[16](64)に         (に)         (51)ニ=格助          in
.[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            comma

*[20](74)形態素解析 (形態素解析) (41)名詞            morphological analysis
.[21](84)は         (は)         (52)ハ=係助          <topic marker>

*[22](86)第一       (第一)       (46)数名詞          the first
.[23](90)の         (の)         (51)ノ=格助          of

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

Figure 3. Segmented Words with Tags

## Morphological Output from QJP

An example of segmented morphemes with morpheme-tags are shown in Figure 2, where 8 nouns ("日本語","単語",etc.) and 2 stems of word ("切"," 置"), marked by '[zJ]', in *kanji* character are recognized using allocation rules and connection table.

The words with part-of-speech tags and morpheme-divisions('-','+') are shown in Figure 3, where a compound noun "切れ目" (the 7th word) is a compound of the morphemes 8-10 ["切"(stem of *shimo-1-dan* verb "切る"), "れ"(*renyou-kei* inflective suffix of *shimo-1-dan* verb; 下一段動詞連用形活用語尾) and "目"(noun)] using a word-constituent rule. In Figure 3, the root forms of inflected words have been derived and are shown in the <>-parentheses, such as "置く" which is the root form (*shuushi-kei*; 終止形) of "置き". These morphemes and words are not in the dictionary.

## 2.2 Syntactic Analysis

### *Kakari-uke* Analysis

Many Japanese syntactic analyses are based on orthodox *bunsetsu*-dependency analysis, called *kakari-uke*[3] analysis(係り受け解析) between *bunsetsu* phrases, where a *bunsetsu*-dependency structure corresponds to a set of *kakari-uke bunsetsu* pairs. We also take this approach because it is intuitive, understandable and easily implemented.

## Simple Treatment of Structural Ambiguities

Structural ambiguities are usually dealt with either by generating all possible structures or by selecting the more preferable ones based on some scoring scheme. Such method usually leads to combinatorial explosions which causes a lot of memory and processing time.

For this problem we have already proposed a substitutional light method[5] in *kakari-uke* analysis. This method extracts all possible *kakari-uke* pairs, and then rather than generate not all or some possible sets of pairs, only one best set of pairs is generated while still retaining all other possible pairs. Thus, instead of generating multiple number of sets, a most-likely set is selected and the application/user is presented with alternative *kakari-uke* pairs at the same time that the selected pairs are presented. If the application/user corrects any alternative *kakari-uke* pairs, the most likely set is re-calculated using retaining possible *kakari-uke* pairs. This means of dealing with structural ambiguities avoids combinatorial explosions and requires far less machine resources.

## Not Using of Semantic Informations

Most methods for analyzing Japanese use case patterns with semantic features for preference selections. However, such analysis techniques using semantic informations are not yet adequate and sometimes actually lead to adverse results[6].

In addition, semantic information must be stored

[3] The relation of *kakari* and *uke* equals to modifier and modifiee.

in the dictionary. This reduces the merit of the very small dictionary achieved in morphological analysis section. We limit the information to morphological/word and syntactic levels [such as the presence of comma(読点), adverbial noun, surface or syntactic similarity[7]] without using semantic information for structural analysis.

## Flow of QJP's Syntactic Analysis

Under these approaches, QJP's syntactic analyser processes words sequence in three steps[Figure 4] each following its own set of rules. First it determines *bunsetsu* features[A] for each *bunsetsu* according to its word constituents. Second it extracts all possible *kakari-uke bunsetsu* pairs [marked by ' ○' in B] based on specific combinations of *bunsetsu* features for each *bunsetsu* pair.

Last, it selects the best *uke-bunsetsu* (modifiee) [marked by ' ●' in C] from possible ones for each *bunsetsu* which is a *kakari-bunsetsu* (modifier), except the last one, because every *bunsetsu* modifies one of the following *bunsetsus*, so the last one has no *uke-bunsetsu*. The default *uke* selection is the nearest possible *uke bunsetsu* and, if necessary, QJP substitutes the selection based on rules comparing the two pairs - the current selected *uke-bunsetsu* and a more distant possible *uke-bunsetsu* for the subject *kakari-bunsetsu*. In Figure 7, some pairs are not the nearest ones. The application/user's *kakari-uke* pairs corrections restarts the selection ; QJP first selects the corrected *kakari-uke* pair(s) [marked by ' ■' in Figure 7] and then re-selects remaining *kakari-uke* pairs.

Figure 4-C and Figure 7 are *kakari-uke* matrices showing the possible pairs and selected pairs. Figure 5 is the output of *kakari-uke* pairs tagged with parts-of-speech and *bunsetsu* features.

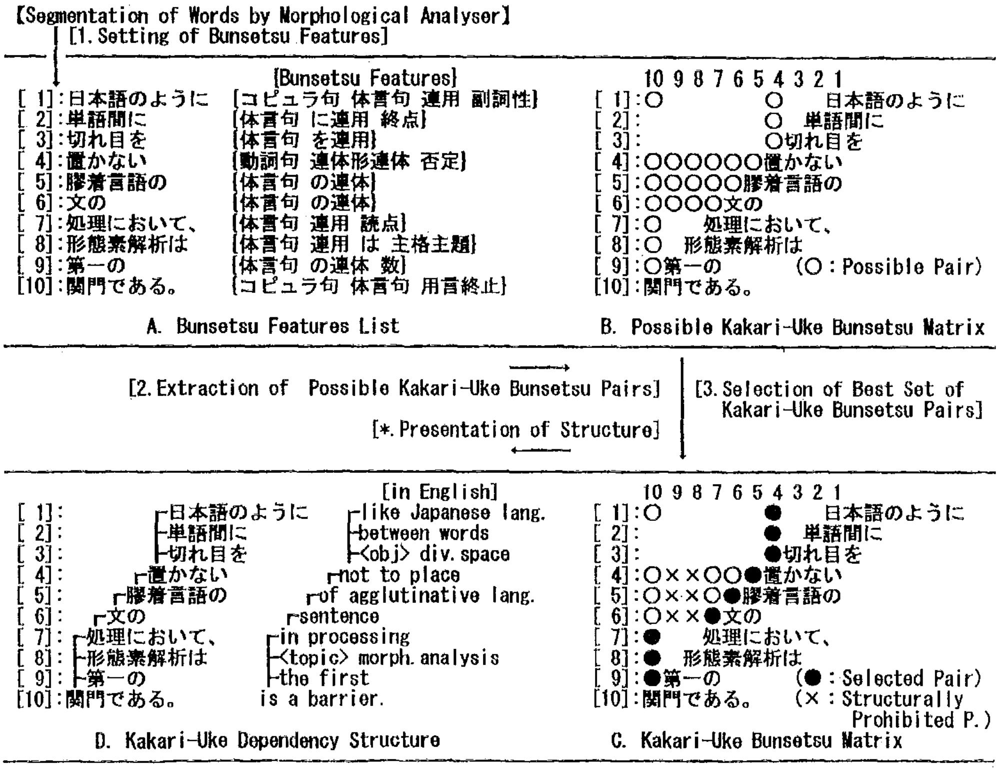

【Segmentation of Words by Morphological Analyser】
[1. Setting of Bunsetsu Features]
{Bunsetsu Features}
[ 1]:日本語のように {コピュラ句 体言句 連用 副詞性}
[ 2]:単語間に {体言句 に連用 終点}
[ 3]:切れ目を {体言句 を連用}
[ 4]:置かない {動詞句 連体形連体 否定}
[ 5]:膠着言語の {体言句 の連体}
[ 6]:文の {体言句 の連体}
[ 7]:処理において、 {体言句 連用 読点}
[ 8]:形態素解析は {体言句 連用 は 主格主題}
[ 9]:第一の {体言句 の連体 数}
[10]:関門である。 {コピュラ句 体言句 用言終止}
A. Bunsetsu Features List
10 9 8 7 6 5 4 3 2 1
[ 1]:○ ○ 日本語のように
[ 2]: ○ 単語間に
[ 3]: ○切れ目を
[ 4]:○○○○○○置かない
[ 5]:○○○○○膠着言語の
[ 6]:○○○○文の
[ 7]:○ 処理において、
[ 8]:○ 形態素解析は
[ 9]:○第一の (○:Possible Pair)
[10]:関門である。
B. Possible Kakari-Uke Bunsetsu Matrix
[2. Extraction of Possible Kakari-Uke Bunsetsu Pairs]
[3. Selection of Best Set of Kakari-Uke Bunsetsu Pairs]
[*. Presentation of Structure]
[in English]
[ 1]: 日本語のように like Japanese lang.
[ 2]: 単語間に between words
[ 3]: 切れ目を <obj> div. space
[ 4]: 置かない not to place
[ 5]: 膠着言語の of agglutinative lang.
[ 6]: 文の sentence
[ 7]: 処理において、 in processing
[ 8]: 形態素解析は <topic> morph. analysis
[ 9]: 第一の the first
[10]:関門である。 is a barrier.
D. Kakari-Uke Dependency Structure
10 9 8 7 6 5 4 3 2 1
[ 1]:○ ● 日本語のように
[ 2]: ● 単語間に
[ 3]: ●切れ目を
[ 4]:○××○○●置かない
[ 5]:○××○●膠着言語の
[ 6]:○××●文の
[ 7]:● 処理において、
[ 8]:● 形態素解析は
[ 9]:●第一の (●:Selected Pair)
[10]:関門である。 (×:Structurally Prohibited P.)
C. Kakari-Uke Bunsetsu Matrix

Figure 4. Flow of Syntactic Kakari-Uke Analysis

```
                 [Segmented Words with Tags]                                          {Bunsetsu Features}               ―<Mod. Type(Kakari:Uke)>→[Mod. No.]

[ 1] [日本語{名詞}の{ノ=格助}ように{ようだ|ヨウダ=ダ用b}]              {コピュラ句 体言句 連用 副詞性}  ――――<連用:動詞句>――→ [4]
[ 2] [単語{名詞}間{形式名詞}に{ニ=格助}]                              {体言句 に連用 終点}             ―――<に連用:動詞句>――→ [4]
[ 3] [切れ目{名詞}を{ヲ=格助}]                                        {体言句 を連用}                  ―――<を連用:動詞句>――→ [4]
[ 4] [置か{置く|動:五カ未a}ない{ない|ナイ=ク体}]                        {動詞句 連体形連体 否定}          ―<連体形連体:体言句>――→ [5]
[ 5] [膠着言語{名詞}の{ノ=格助}]                                      {体言句 の連体}                  ―――<の連体:体言句>――→ [6]
[ 6] [文{名詞}の{ノ=格助}]                                            {体言句 の連体}                  ―――<の連体:体言句>――→ [7]
[ 7] [処理{名詞}に{ニ=格助}おい{おく|オク=五カ用b}て{テ=接助}、{読点}]  {体言句 連用 読点}               ――――<連用:コピュラ句>→[10]
[ 8] [形態素解析{名詞}は{ハ=係助}]                                    {体言句 連用 は 主格主題}          ――<主格主題:コピュラ句>→[10]
[ 9] [第一{数名詞}の{ノ=格助}]                                        {体言句 の連体 数}               ―――<の連体:体言句>――→[10]
[10] [関門{名詞}で{だ|ダ=用c}ある{ある|アル=五ラ終}。{句点}]            {コピュラ句 体言句 用言終止}
```

Figure 5. Kakari-Uke Pairs with Tags

# 3 QJP

## 3.1 Implemented Software

QJP currently is implemented in the C language as both a QJP library and an interactive/batch console application, QJP workbench. They have been implemented on DOS/PC and UNIX/Sun workstation.

QJP's dictionary consists of 4 files whose total size is less than 50KB and which contain about 5 thousands morphemes. QJP requires another control table file for the compressed 533×533 morpheme-POS connection table, the table for the allocation rules, the dictionary file indices and others, which is at most 35KB. 6 sets of morphological rules and 4 sets of syntactical rules [Table 2] are embedded in the form of *if-then rule* in C functions. The size of the workbench execution file on DOS is about 185KB. The total size (executables and dictionaries) is much less than 300KB[DOS] which is quite small and portable as a natural language analyser.

## 3.2 Analysis Experiment

QJP performances were measured for the QJP workbench using two sentence test-sets : 1 [241 sentences, average length 24.1 words/sentence] and 2 [210 sentences, average length 29.5 words/sentence].

### Execution Performance

About 260KB of memory are required on DOS and 500KB on UNIX. With this amount of memory QJP can process a very long sentence, such as 100-word sentence [Figure 6].

The analysis speed is 80 to 150 words/sec on an 80486/25MHz PC and 700 to 800 words/sec on a Sun-SS20. A 100-word sentence can be analyzed in less than 1 second on PC. Figure 8 shows the relationship of processing time to sentence length. Syntactic processing time is on the order of the square of the sentence length. But its coefficient is so small that the total processing time increases linearly in the range of actual long sentences.

Table 2. Linguistic data and Rules

| | |
|---|---|
| Morphological Analyser | |
| [D] dictionary : ≒3500 entries, ≒5000 morphemes | |
| [P] morpheme/word Part-of-Speech | : 533/49 POSs |
| [T] connection table | : 533×533 |
| ( └── [R] connection source-rules | : ≒300 rules) |
| [R] char.sequence extraction rules | : ≒ 20 rules |
| [R] morpheme-POS allocation rules | : 14 rules |
| [R] morpheme-POS disambiguation rules | : ≒ 50 rules |
| [R] word-constituent rules | : ≒ 60 rules |
| [R] bunsetsu head exceptional rules | : ≒ 20 rules |
| [R] auxiliary functional verb rules | : ≒ 15 rules |
| Syntactical Analyser | |
| [F] bunsetsu features | : 68 features |
| [R] bunsetsu features setting rules | : ≒ 80 rules |
| [R] kakari-uke pair extracting rules | : ≒ 20 rules |
| [R] kakari-uke pair exceptional rules | : ≒ 40 rules |
| [R] kakari-uke failure recovery rules | : ≒ 4 rules |

※[D]:dictionary [T]:table [R]:rule [P]:POS [F]:feature

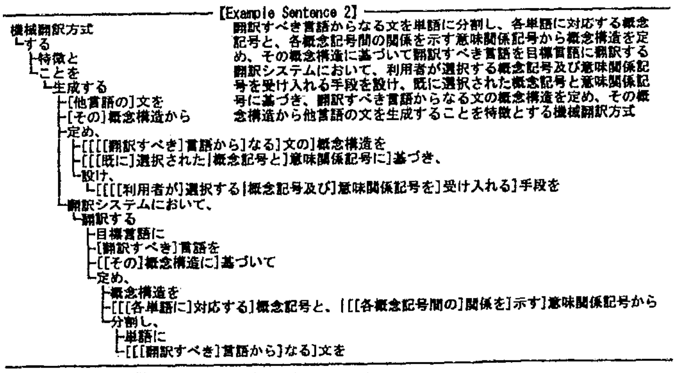

[Example Sentence 2]
翻訳すべき言語からなる文を単語に分割し、各単語に対応する概念記号と、各概念記号間の関係を示す意味関係記号から概念構造を定め、その概念構造に基づいて翻訳すべき言語を目標言語に翻訳する翻訳システムにおいて、利用者が選択する概念記号及び意味関係記号を受け入れる手段を設け、既に選択された概念記号と意味関係記号に基づき、翻訳すべき言語からなる文の概念構造を定め、その概念構造から他言語の文を生成することを特徴とする機械翻訳方式
機械翻訳方式

Figure 6. Analyzed Syntactic Structure for a 50-bunsetsu senence

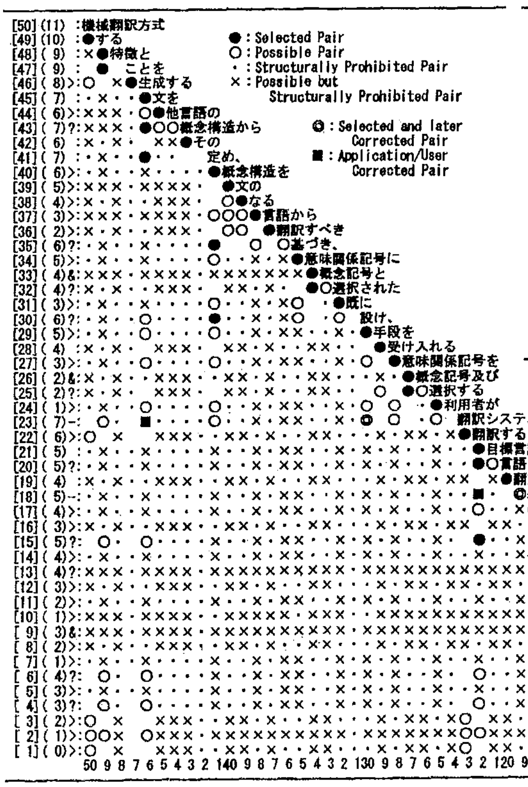

●: Selected Pair
O: Possible Pair
・: Structurally Prohibited Pair
×: Possible but Structurally Prohibited Pair
◎: Selected and later Corrected Pair
■: Application/User Corrected Pair

Figure 7. Kakari-Uke Matrix for a 50-bunsetsu senence

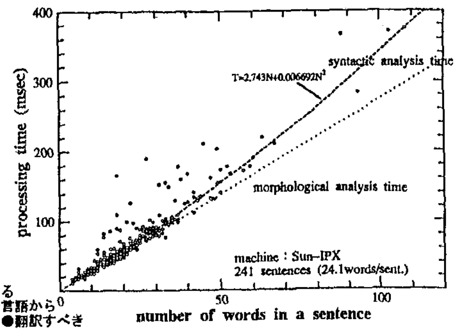


Figure 8. Processing Time vs. Sentence Length

### Analysis Performance

We used test-sets 1 and 2 for tuning and blind-test, respectively. For test-sets 1 and 2, the accuracy of analyzed morphemes/words is 99.3/99.3% and 95.7/96.1%, the accuracy of analyzed *uke bunsetsus* for each *bunsetsu* except the last one is 95.1% and 90.5%, and the accuracy of set of *kakari-uke* pairs in a sentence is 71.0% and 43.8%, respectively.

For sentences which have lengths of 3 to 15-bunsetsus and are morphologically analyzed correctly, the accuracy of analyzed *uke bunsetsus* for each *bunsetsu* is 97.3% and 93.6%, and the accuracy for sets of *kakari-uke* pairs in a sentence is 82.9% and 70.5%, respectively.

### Comparison

There are no public data for the performance of other Japanese analysers, so comparison is difficult. But not only the size of files but also the performance figures for memory and speed of QJP are thought to be more than ten times better than those of existing Japanese analysers[4]. As for analysis accuracy, the morphological accuracy is a little lower than that of the existing Japanese morphological analysers using large scale dictionaries, but the syntactic analysis accuracy is thought to be no worse than that of the existing Japanese syntactic analysers.

## 4 Conclusions

We have designed and implemented QJP for the purpose of readily and easily applicable morphological and syntactic analyser for Japanese. The design strategies are based on 1) the morphological analysis based on character-types and functional words to reduce the size of dictionary, and 2) the syntactic analysis by simple treatment of structural ambiguities and ignoring semantic information to lighten processing.

QJP, as implemented, is portable, quick and robust. All files needed for execution including dictionary total less than 300KB on DOS. Even on a slow PC a 100-word sentence can be analyzed in less than 1 second using a small amount of memory. This performances is thought to be quite excellent. The analysis accuracy is comparable to that of other existing analysers. No dictionary maintenance is necessary for new terms.

The functions of QJP are implemented as a QJP library and a QJP workbench. We have already utilized QJP for keyword extraction, natural language query and text reading support functions[9] and are planning further applications, such as information retrieval system. Others use QJP functions for other purposes, such as linguistic data extraction.

QJP currently doesn't segment compound kanji words of Chinese-origin and leaves this segmentation to the application. In the future, we plan to realize such a segmenting function using on statistical data[10] and affixes[2].

## References

[1] Yoshiyuki SAKAMOTO: An Automatic Segmentation for Japanese Text, Mathematical Linguistics, Vol.11, No.6, 1978 (in Japanese).

[2] Makoto NAGAO, Jun-ichi TSUJII, Akira YAMAGAMI, Shuji TATEBE: Data-Structure of a Large Japanese Dictionary and Morphological Analysis by Using It, Journal of Information Processing Society of Japan, Vol.19 No.6, pp.514-521, 1978 (in Japanese).

[3] Masayuki KAMEDA: A Quick Japanese Parser, Information Processing Society of Japan SIG Notes, 94-NL-4, 1993 (in Japanese).

[4] Masayuki KAMEDA: A Portable & Quick Japanese Processing Tool : QJP, Proc. of the 1st Annual Meeting of the Association for Natural Language Processing, 1995 (in Japanese).

[5] Masayuki KAMEDA, Shin ISHII, Hideo ITO: Interactive Disambiguation in a Japanese Analysis System, Information Processing Society of Japan SIG Notes, 84-NL-2, 1991 (in Japanese).

[6] 渕武志, 米澤明憲, 田中康仁: 係り受けの曖昧性解消における共起データの効果の測定, EDR 電子化辞書利用シンポジウム 論文集, 1995 (in Japanese).

[7] Sadao KUROHASHI, Makoto NAGAO: A Method for Analyzing Conjunctive Structures in Japanese, Transactions of Information Processing Society of Japan, Vol.33 No.8, pp.1022-1031, 1992 (in Japanese).

[8] Sadao KUROHASHI, Toshihisa NAKAMURA, Yuji MATSUMOTO and Makoto NAGAO : Improvement of Japanese Morphological Analyser JUMAN, Proc. of the International Workshop on Sharable Natural Language Resources, pp.22-28, 1994.

[9] Masayuki KAMEDA: Support functions for Reading Japanese text, Information Processing Society of Japan SIG Notes, 110-NL-9, 1995 (in Japanese).

[10] Koichi TAKEDA, Tetsunosuke FUJISAKI: Automatic Decomposition of Kanji Compound Words Using Stochastic Estimation, Transactions of Information Processing Society of Japan, Vol.28 No.9, pp.952-961, 1987 (in Japanese).

### Example Sentences

1 Tohru HISAMITSU: Proc.of the 42th Meeting of Information Processing Society of Japan, 1991.

2 Japanese Laid Open Patent No.60-200368, 1985.