Regular Paper

An Accurate Morphological Analysis and Proper Name Identification for Japanese Text Processing

TSUYOSHI KITANI[†,††] and TERUKO MITAMURA[†]

This paper describes a Japanese preprocessor used for syntactic and semantic parsing. It consists of three major components: (1) a morphological analyzer called MAJESTY (Morphological Analyzer for Japanese Text Analysis), (2) a proper name identification and grouping program, and (3) a format conversion program for an input to Tomita's generalized LR parser. To enable the parser to perform efficiently, the original morphological analyzer was modified to disambiguate its output when multiple possibilities for segmentations and parts of speech were found, and to pack ambiguous segments locally in the output. The grouping program identifies several segments forming one concept, which is often the case with proper names, and puts them together to provide a meaningful set of segments for the parser. The grouped segments are finally converted into a Lisp readable format and fed into the parser. Tested on financial news articles, the preprocessor successfully segmented text and tagged parts of speech with a greater than 98% accuracy. Company names have been identified with over 80% in both recall and precision. Person and place names have also been recognized with over 90% accuracy. The preprocessor has been successfully integrated into the SHOGUN and TEXTRACT information extraction systems which process texts in the TIPSTER domains of corporate joint ventures and microelectronics.

1. Introduction

In this paper, a Japanese preprocessor, which was developed to process Japanese news articles, is described from the point of view of parser efficiency. In the Center for Machine Translation, Carnegie Mellon University, Japanese morphology rules have been written in the same formalism as syntactic grammar rules for narrow domain applications, such as the Doctor-Patient machine translation system.[1] Both kinds of rules are compiled into an LR table used by Tomita's generalized LR parser to parse the text in a character-based mode.[2] The advantage of this architecture is that the morphology rules can be applied during syntactic and semantic processing, which enables the parser to perform accurate morphological analysis using syntactic and semantic knowledge. However, when the domain becomes more general, the increase in lexicon size makes the LR table so huge that character-based parsing is no longer practical in terms of processing speed or memory size. Moreover, for other parsers designed to process segmented text, it is essential to provide segmented input in order to analyze Japanese text, since no space characters are placed between words. Thus, for general domain applications, it is necessary to run the morphological analyzer separately from the parser as a preprocessor. The morphological analyzer, however, must be accurate and fast.

The original Japanese morphological analyzer has been used as a major component in a Japanese text proofreading system[3] and in an OCR post-processing system.[4] It segments text, tags parts of speech, and even gives the pronunciation of a word in Roman letters. Although it has proved accurate enough for those types of applications, it is necessary to consider the following issues for efficient parser performance in other applications: (1) Disambiguation of multiple segmentations and parts of speech, ranking both in a likely order; (2) Grouping several morphemes that form a concept to provide meaningful sets of segments; (3) Representation of ambiguous output in a format that a parser can process efficiently.

2. Problem Definition

2.1 Ambiguity

Two types of ambiguity are generated in a

† Center for Machine Translation, Carnegie Mellon University
†† Visiting researcher from NTT Data Communications Systems Corp.

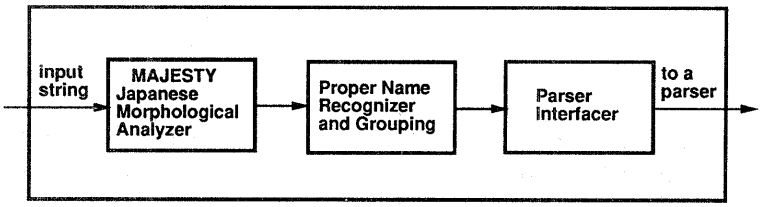


Fig. 1 Module structure of the Japanese preprocessor.

Japanese morphological analyzer: part of speech ambiguity and segmentation ambiguity. The segmentation ambiguity is peculiar to languages written without space characters between words. The following example shows both types of ambiguity,

1. 米国[NOUN-PLACE] 東[NOUN]
   (America) (east)
   海岸[NOUN]
   (coast)
2. 米国[NOUN-PLACE]
   (America)
   東海[NOUN-PLACE] 岸[NOUN]
   (Tokai) (coast)
3. 米[NOUN-PLACE, NOUN]
   (America, rice)
   国東[NOUN-PLACE] 海岸[NOUN]
   (Kunisaki) (coast)

where a part of speech is shown in "[ ]" and a corresponding English word is shown in "( )". For proper names, an attribute such as COMPANY, PLACE, FAMNAME (family name), and FSTNAME (first name) follows the part of speech in the square brackets.

Without semantic information, these ambiguities are inevitable during preprocessing, but it is desirable to limit them as much as possible. When ambiguities exist, the parser should expect ambiguous possibilities to be ranked in a likely order. Furthermore, ambiguities should be packed locally so that the parser can easily compare the differences between several possibilities.

2.2 Segmentation granularity

A Japanese morphological analyzer usually segments a text into the smallest pieces of words possible due to the necessity of recognizing individual words. On the other hand, a parser has to put words together into meaningful sets of segments during syntactic and semantic parsing. Therefore, grouping meaningful sets of segments with a preprocessor helps reduce the work of the parser. Such a case is often seen in a proper name which is formed from several words. A Japanese morphological analyzer, however, usually fails to recognize those words as a compound word unless it is defined in the dictionary. Here is an example of a segmented company name recognized by a Japanese morphological analyzer.

日本[NOUN-PLACE] 放送[NOUN]
(Japan) (broadcasting)
協会[NOUN]
(association)

It is desirable to have the whole string grouped as a [NOUN-COMPANY] for the parser.

Since proper names can be one of the most important pieces of information in analyzing a topic in a text, it is important to identify them precisely. They can also be used as key words in database generation and for data retrieval from databases.

3. The Solution

The Japanese preprocessor is comprised of three major modules as shown in **Fig. 1**. In the Japanese preprocessor, solutions to the problems described in Section 2 are based on heuristic knowledge collected from corpus data. Corporate joint venture newspaper articles were chosen for the corpus.*

3.1 Morphological analyzer—MAJESTY

3.1.1 The basic algorithm and dictionary

MAJESTY adapted a general algorithm for a morphological analysis which is widely used in many Japanese morphological analyzers.[5] The algorithm can be broken into 4 steps: (1) dividing an input string where delimiters such as "、"

* Corpus data was provided from ARPA to the Center for Machine Translation, Carnegie Mellon University, for the research of the TIPSTER information extraction project.

and "。" appear, (2) looking up in the dictionary all possible substrings derived from the divided strings, (3) checking the logical connection between adjacent words based on the part of speech, inflection, and word length, and (4) choosing connection-approved words and creating output paths in which the least number of content words exist.

MAJESTY's main dictionary contains about 91,500 words consisting of company, person and place names as well as common words as shown in **Table 1**. Words are categorized into 54 parts of speech necessary for accurate segmentation. Additional details about the dictionary and tables are explained in Ref. 4).

3.1.2 Disambiguation and Ambiguity Packing

Based on the observation of Japanese language phenomena, ambiguous segmentations and parts of speech were categorized into the following three types.

- TYPE-I: one possibility only was almost always correct
  (Example 1) "3台"
  The part of speech of "台(dai)" should be analyzed as a suffix for numbers indicating unit, instead of as a regular noun indicating a table, since it appears right after a numeric character.
  (Example 2) "...があり、"
  "あり(ari)" (exist) should be segmented as "あ" [VERB] and "り" [INFLECTION], instead of "あり" [NOUN] meaning an ant, since the word "ant" is not likely to appear in the joint venture and microelectronics domains.
- TYPE-II: one possibility only was correct most of the time
  (Example 3) "...集め、"
  "集め(atsume)" (gather) is likely to be a verb rather than a noun, since a non-ending verb form is preferable to a noun before a Japanese comma ("、").
  (Example 4) "...走り。"
  "走り(hashiri)" [NOUN] (a run) is a preferred segmentation rather than "走" [VERB] and "り" [INFLECTION], since a noun is preferable to a non-ending verb form before a Japanese period "。".
- TYPE-III: a correct possibility depended on the context
  Both segmentation and part of speech ambiguities can be seen in the example of "米国東海岸(beikoku higashi kaigan)" shown in Section 2.1.

Table 1 Number of content words defined in MAJESTY's dictionary.

Category	Number of words
Common words	72,332
Company names	3,175
Family names	4,314
First names	3,354
Place names	8,291

Common features of segmentations and parts of speech, responsible for one possibility being selected among several possibilities, were collected by running MAJESTY over the corpus. TYPE-I ambiguities were captured by 4 rules for segmentation selection and 4 rules for part of speech selection. TYPE-II common features were captured by 24 rules for segmentation preference and 11 rules for part of speech preference.

While a morphological analysis is taking place, MAJESTY keeps all possible paths chained by connection-approved segments and parts of speech slightly different from each other. At the last stage of the process, MAJESTY locates segmentations and parts of speech that are different from each other. MAJESTY then packs them locally and creates one path only as a whole. The packing is done in such a way that it generates a new path where an ambiguous segmentation gets nested or overlaps. **Figure 2** shows an internal representation of ambiguous paths where W_{is} represents a sth segment of path i, and $H_{jm,n}$ means an nth possible part of speech at mth segment of path j.

With the Fig. 2 representation, an example of a selection rule can be written as:

IF
  ($H_{i1,1}$=NOUN) & ($H_{j1,1}$=VERB) &
  (W_{b1}="、")
THEN
  delete(j, *OTHERS*).

In this rule, common features of segmentations and parts of speech are captured by the appearance of a NOUN and a VERB in different segmentation paths followed by "、" (a Japanese comma). The rule is defined to choose a path including a VERB rather than a NOUN because

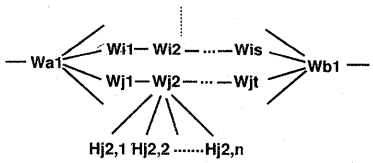


Fig. 2 Internal representation of ambiguous segmentations and parts of speech.

a VERB is strongly desired before "、". The function "delete" takes a path number and a flag as parameters. The flag can be either "OTHERS" or "ITSELF", and makes *delete*(*j*, *OTHERS*) delete all paths except path *j*. Another function, "move", is prepared to reorder a path into "TOP" or "BOTTOM" among possible paths.☆

An advantage of this heuristic approach is the capability of selecting ambiguities desired by a parser. In fact, some TYPE-II preference rules give a priority to "extended function words" proposed by Shudo[6] for efficient syntactic and semantic parsing. These rules reduce many ambiguities commonly seen in the output generated by a morphological analyzer.[7]

A Japanese morphological analyzer called JUMAN,[8] developed by KYOTO university, was reported to achieve high accuracy by incorporating a dictionary provided by the Institute for New Generation Computer Technology (ICOT).☆☆ However, JUMAN simply generates all paths composed of possible segmentations and parts of speech. In comparison, the feature of local ambiguity packing provided by MAJESTY will be favorable to the parser, since the differences among several possibilities can be easily located and compared by the parser.

3.1.3 Unknown word detection

In addition to the proper treatment of ambiguity, detecting unknown words also plays an important role in achieving high accuracy in a morphological analyzer.[9] A basic method for detecting unknown words in an input string is to locate a substring whose words are not defined in the dictionary. An unknown word can also be detected when a logical connection between adjacent words is disapproved, even though the word is found in the dictionary. However, this method does not always work properly, since a word can often be separated into small pieces of words. For example, the unknown word "エアロスペース(aerospace)" can be divided as "エア(air [NOUN])", "ロス([UNKNOWN])", and "ペース(pace [NOUN])". To deal with this undesirable behavior, the following heuristic rules were introduced on top of the basic method to identify unknown words.

- identify a whole Katakana or alphabetical string as an unknown word, even if only a fragment of it is unknown, and
- disapprove a logical connection between adjacent words both consisting of one character, unless they are person names.

These rules can be regarded as a simple implementation of an unknown word detection algorithm published by Yoshimura.[7]

3.2 Proper name identification and grouping

In Japanese news articles, company, person, and place names are usually expressed in the following pattern.

[prefix [appositive]] PROPER-NAME ["("explanation")"]

Fields surrounded by "[ ]" can be omitted. Parentheses always appear in the explanation field. As shown in **Table 2**, prefixes, appositives and suffixes are extracted from the corpus data for each kind of proper name.

3.2.1 Overview of the identification algorithm

The idea of identifying company names in English texts using company name suffixes was published by Rau.[10] We enhanced it to identify not only company names but also person names and place names.[11] Proper names are identified in the following ways using segmented texts tagged with parts of speech for input.

1. Detection by MAJESTY
   In news articles, familiar proper names are sometimes expressed by themselves without accompanying suffixes or prefixes. They are expected to be identified in MAJESTY by the dictionary lookup.
2. Detection from prefixes and suffixes
   Unfamiliar proper names are usually accompanied by prefixes or suffixes to identify them. Therefore, prefixes and suffixes

☆ Although a path is moved only to either "TOP" or "BOTTOM", a favorable possibility is almost always ranked at the top. This is explained in a later section.

☆☆ Announced by Professor Matsumoto at a summer tutorial held by the Japan Software Science in August, 1992.

Table 2 An example of prefixes, appositives and suffixes.

	Prefixes	Appositives	Suffixes
Company	"大手" [14] (leading) "企業" (company)	"の" [2] (of) "である" (being)	"社" [52] (Inc.) "グループ" (group)
Person	"社長" [11] (president) "取締役" (director)	"・" [2] "、" (,)	"氏" [17] (Mr.) "社長" (president)
Place	"本社" [1] (headquarters)	"・" [2] "、" (,)	"市" [7] (city) "州" (state)

The number of defined words is shown in [ ].

are used as identification keys of unfamiliar proper names. A search for a proper name originates at either a prefix or suffix. However, when either of these is missing in the text, the pattern search must stop at the other end of the proper name where a prefix or suffix does not exist. Since proper names are not usually composed of Hiragana or symbolic characters,* the search is designed to stop at a segment in which those kinds of character appear. The termination condition was validated empirically by hundreds of proper name occurrences in the corpus.

3. Detection of abbreviations
Proper names can be abbreviated when they occur more than once. To identify such occurrences, possible abbreviations for words identified previously in the news article are generated by removing substrings that are identical to suffixes, and by searching for alphabetical words in the proper name and the explanation fields. The generated substrings are stored in the form of regular expressions and are again matched against unknown words and proper names in the article which are likely to be proper name abbreviations.

3.2.2 Detailed description of the identification algorithm

For the word position i, let W_i be a word and H_i be a part of speech, and let P_p, P_a, P_f, P_t, P_s and P_e be the position of a prefix, the position of an appositive, the starting position of a proper name, the ending position of the proper name, the position of a suffix, and the ending position of the explanation field, respectively. The position originates at the first segment of input. Position "0" means that an item of interest is not found in the input. Also S_i is defined as an item pair $\langle P_i, A_i \rangle$ which stores all the positions of P_p, P_a, P_f, P_t, P_s and P_e as P_i, and the kind of the proper name as A_i. When the number of input words is n, the detailed description of the proper name identification and grouping algorithm is as described below. A simple example showing the identification process is illustrated in **Fig. 3**.

1. STEP 1: Create regular expressions.
   (1) $R_p \leftarrow$ Regular expressions of all predefined prefixes
   (2) $R_a \leftarrow$ Regular expressions of all predefined appositives
   (3) $R_s \leftarrow$ Regular expressions of all predefined suffixes
   (4) $R_d \leftarrow$ Null string for initialization
   (5) $j \leftarrow 1$
2. STEP 2: Match pattern against all words W_i $(i=1, 2, \cdots, n)$.
   (1) Search for a proper name identified by MAJESTY
   IF W_i is a proper name identified by MAJESTY, THEN
   i. $P_j \leftarrow (0, 0, i, i, 0, 0)$
   ii. $S_j \leftarrow \langle P_j, A_i \rangle$
   iii. $j \leftarrow j+1$, GOTO STEP 2
   (2) Search for a proper name from a prefix
   IF $W_i \tilde{} R_p$ ("~" is an operator searching for patterns that match the regular expressions), THEN
   i. Search for a proper name pattern and decide P_a, P_f, P_t, P_s, P_e
   ii. $P_j \leftarrow (i, P_a, P_f, P_t, P_s, P_e)$
   iii. $S_j \leftarrow \langle P_j, A_i \rangle$
   iv. $j \leftarrow j+1$
   (3) Search for a proper name from a suffix
   IF $W_i \tilde{} R_s$, THEN
   i. Search for a proper name pattern and decide P_p, P_a, P_f, P_t, P_e
   ii. $P_j \leftarrow (P_p, P_a, P_f, P_t, i, P_e)$
   iii. $S_j \leftarrow \langle P_j, A_i \rangle$
   iv. $j \leftarrow j+1$

* Symbolic characters such as "「", "」", and "・" are exceptions, since they are often a part of proper names in news articles.

Output from MAJESTY

```
日本銀行 [NOUN-COMPANY]  と [P]  証券 [NOUN]  大手 [NOUN]  の [P]
(Bank of Japan)          (and)  (a major security company)

鈴木 [NOUN-PERSON]  証券 [NOUN]  は [P]  、 [SYMBOL]  鈴木 [NOUN-PERSON]  の [P]
(Suzuki)            (Securities)                      (Suzuki)
```

Company name identification process

```
STEP 1: Create regular expressions (initialization)
    Rp = "大手" | "企業" | ・・・      (prefixes)
    Ra = "の" | "である" | ・・・      (appositives)
    Rs = "証券" | "銀行" | "社"・・・ (suffixes)
    Rd = ""

STEP 2: Identify company names using the regular expressions
    日本銀行 (identified by MAJESTY)
    鈴木証券 (identified by prefix "大手")
    鈴木証券 (identified by suffix "証券")

STEP 3: Choose patterns
 (1) Remove overlapping patterns
    "鈴木証券" identified by prefix "大手" is removed.
 (2) Create regular expressions to detect abbreviations
    Rd = "日本" | "鈴木"  (abbreviations)

STEP 4: Identify abbreviations
    鈴木 (identified by Rd)

STEP 5: Generate company name output
    日本工業
    鈴木証券
    鈴木
```

Fig. 3 An example of company name identification process.

3. STEP 3: Choose correct patterns from all matched patterns P_j $(j=1, 2, \cdots, jmax$, where $jmax$ is the number of proper names identified by STEP 2), and create regular expressions to be used to match abbreviated proper names in STEP 4.
   (1) IF $(A_j=A_{j+1})$ & $(max(P_{tj}, P_{sj}) \geq min(P_{pj+1}, P_{aj+1}, P_{fj+1}))$ & $((max(P_{tj}, P_{sj}) - min(P_{pj}, P_{aj}, P_{fj})) \leq (max(P_{tj+1}, P_{sj+1}) - min(P_{pj+1}, P_{aj+1}, P_{fj+1})))$, THEN
      i. $S_j \leftarrow \langle 0, 0 \rangle$
      where $max(P_{tj}, P_{sj})$ means the maximum value of P_t and P_s in P_j, and $min(P_{pj}, P_{aj}, P_{fj})$ means the minimum value of P_p, P_a and P_f in P_j. Thus, a narrower overlapping pattern is deleted if it is matched by the same kind of proper name.
   (2) ELSE
      i. $R_d \leftarrow R_d | (W_i \rightharpoonup R_s)$
      A "$\rightharpoonup$" operator removes substrings that are found in both strings to be compared. In this case, substrings in the word W_i that are identical with suffixes R_s are removed. Then, the leftover string of W_i is added to the regular expression R_d. An operator "|" means to add a string in the form of regular expressions.
      ii. $R_d \leftarrow R_d | W_e$
      where W_e is an alphabetical string appearing in the proper name and explanation fields.
4. STEP 4: Match pattern to identify abbreviated expressions against unknown words and proper names in W_i $(i=1, 2, \cdots, n)$.
   (1) IF W_i is an unknown word or a proper name and $W_i \sim R_d$, THEN
      i. $P_j \leftarrow (0, 0, i, i, 0, 0)$
      ii. $S_j \leftarrow \langle P_j, A_i \rangle$
      iii. $jmax \leftarrow jmax+1$

5. STEP 5: Create output for all matched patterns P_j $(j=1, 2, \cdots, jmax)$.
   (1) IF $S_j \neq \langle 0, 0 \rangle$, THEN
      i. Group words from P_f to P_s in P_j and print them out

3. 2. 3 Other grouped segments

In addition to proper names, numeric expressions (including temporal and monetary expressions) are also grouped in the manner described in 3. 2. 2. Another unit to be grouped is a root word followed by its affixes and function words such as an inflection and auxiliary verb. This grouping is particularly important for a parser to recognize a set of words containing modal and tense information.

3. 2. 4 Output format

Figure 4 shows an example of the grouped output with SGML (Standard Generalized Markup Language)[12] tags used to distinguish fields. Each segment has a word string, a part of speech and a pronunciation in Roman letters surrounded by the ⟨TOK⟩ and ⟨/TOK⟩ tags. Ambiguous segments are packed between the ⟨OR⟩ and ⟨/OR⟩ tags while accommodating segments in each possible path between the ⟨DIF⟩ and ⟨/DIF⟩ tags. Ambiguous parts of speech are simply listed between the ⟨POS⟩ and ⟨/POS⟩ tags.

Elements of ⟨GRP-n⟩, ⟨GTOK⟩ and ⟨GPAR⟩ are added to MAJESTY's output by the proper name identification and grouping program. Grouped segments can be recognized by the ⟨GTOK⟩ tag. The part of speech for the grouped segments is given by the ⟨GPAR⟩ tag.

```
<GRP-1>
<GTOK> ジョーンズ社 </GTOK>
<GPAR>COMPANY</GPAR>
    <TOK><FTOK> ジョーンズ </FTOK><FPOS>NOUN-FAMNAME NOUN-COMPANY</FPOS><FROM>jo-Nzu</FROM></TOK>
    <TOK><FTOK> 社 </FTOK><FPOS>SUFFIX-NOUN NOUN</FPOS><FROM>sha</FROM></TOK>
</GRP-1>
<TOK><FTOK> は </FTOK><FPOS>P</FPOS><FROM>wa</FROM></TOK>
<OR><DIF>
<TOK><FTOK> さらに </FTOK><FPOS>ADV</FPOS><FROM>sarani</FROM></TOK>
</DIF><DIF>
<TOK><FTOK> さら </FTOK><FPOS>NOUN</FPOS><FROM>sara</FROM></TOK>
<TOK><FTOK> に </FTOK><FPOS>P</FPOS><FROM>ni</FROM></TOK>
</DIF></OR>
<GRP-1>
<GTOK> 3割 </GTOK>
<GPAR>NUMBER</GPAR>
     <TOK><FTOK> 3 </FTOK><FPOS>NUM</FPOS><FROM>3</FROM></TOK>
     <TOK><FTOK> 割 </FTOK><FPOS>SUFFIX-NUM NOUN</FPOS><FROM>wari</FROM></TOK>
</GRP-1>
<TOK><FTOK> も </FTOK><FPOS>P</FPOS><FROM>mo</FROM></TOK>
<GRP-1>
<GTOK> 増資した </GTOK>
<GPAR>BUNSETSU</GPAR>
     <TOK><FTOK> 増資 </FTOK><FPOS>SAHEN</FPOS><FROM>zoushi</FROM></TOK>
     <TOK><FTOK> し </FTOK><FPOS>INFLECTION-RENYO4</FPOS><FROM>shi</FROM></TOK>
     <TOK><FTOK> た </FTOK><FPOS>AUX-PAST</FPOS><FROM>ta</FROM></TOK>
</GRP-1>
```

Fig. 4 Example of a grouped output.

```
((string ジョーンズ社) (POS NOUN) (SUBPOS COMPANY)
 (group
    ((string ジョーンズ) (POS NOUN) (SUBPOS FAMNAME) (ROMA jo-Nzu))
    ((string 社) (POS SUFFIX) (SUBPOS NOUN) (ROMA sha))
))
((string は) (POS P) (ROMA wa))
((string さらに) (POS ADV) (ROMA sarani))
((string 3割) (POS SUFFIX-NUM) (SUBPOS NUMBER)
 (group
    ((string 3) (POS NUM) (ROMA 3))
    ((string 割) (POS SUFFIX-NUM) (ROMA wari))
))
((string も) (POS P) (ROMA mo))
((string 増資した) (HEAD 増資) (POS VERB) (ROMA zoushi) (TYPE  RENYO) (TENSE PAST))
```

Fig. 5 Preprocessor's output in a Lisp readable format.

As shown in **Fig. 5**, it is then converted into a Lisp readable format* as the output of the preprocessor, which is currently used as an input to Tomita's generalized LR parser.

4. Evaluation

About a thousand news articles were examined in developing the disambiguation rules for MAJESTY. The same articles were also used to develop the proper name identification and grouping algorithm. A different set of articles was used for the subsequent evaluation.

4. 1 Disambiguation rules

Thirty-one articles comprising 9,451 words of text were used to evaluate the disambiguation rules. There were 320 unknown words in the test corpus. The evaluation results are shown in **Fig. 6**.

The selection rules for reducing the number of ambiguities worked for all the 58 ambiguous segments and for all the 164 ambiguous parts of speech that were categorized as having only one correct possibility (TYPE-I described in Section 3. 1. 2). The reordering rules, which move a preferred possibility to its most likely place, worked in 120 cases out of the 152 ambiguous TYPE-II segments, and in 368 cases out of the 371 ambiguous TYPE-II parts of speech. The reordering rules, however, did not entirely rank possibilities in the likely order, since the reordering was done by moving a possibility into the most or the least likely place. Based on the fact that the average number of multiple segmentations and parts of speech were 2. 1 and 2. 2, respectively, they were practically reordered correctly with few errors.

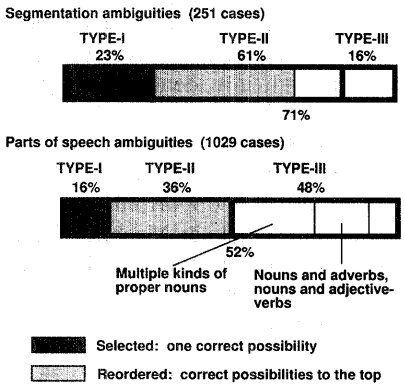


Fig. 6 Results of applied disambiguation rules.

In TYPE-III parts of speech ambiguities, combinations of nouns and adverbs, or nouns and adjective-verbs accounted for one-third of the total. Since no particular distinction between these cases could be recognized in adjacent words at the morphological analysis level, the correct part of speech must be selected during syntactic and semantic processing.

Regarding the rest of the cases that were not reordered, no common features of segmentations or parts of speech among the possibilities could be found. For those TYPE-II and TYPE-III ambiguities, reordering them based on word frequency obtained from the corpus data will be effective.

MAJESTY's accuracy is defined as the number of correct segments in the output compared to the number of correct segments existing in the text. Only the most likely segmentation and part of speech were taken into account for the accuracy measurement. When only the unknown word detection algorithm was implemented in the original morphological analyzer, its accuracy was nevertheless 93.9%. However, the addition of disambiguation rules successfully improved MAJESTY's accuracy to 98.2%.

4. 2 Proper name identification and grouping

The proper name identification and grouping algorithm was tested with corpus data which included 312 company names, 72 person names, 325 place names, and 167 numeric expressions, each identified by a human indexer. The identification results are shown in **Table 3**. The evaluation measures were *recall* (how much information was extracted) and *precision* (how much of the extracted information was correct).

Among the various kinds of proper names, person name identification results showed the highest recall and precision. This was due to the fact that a person-name suffix, which became a key word for the pattern matcher, was usually found in a news article. On the other hand, company names were not as well identified as expected. The absence of company-name prefixes and suffixes became the major cause of low accuracy as few non-Japanese company names were defined in the MAJESTY dictio-

* Figure 5 shows the most likely segmentation and part of speech only for the simplicity.

Table 3 Proper name identification and grouping results.

	Company	Person	Place	Number
Recall	84.3%	93.1%	92.6%	85.0%
Precision	81.4%	98.6%	96.8%	99.3%

Table 4 Types of matches on company names.

	MAJESTY	prefix, suffix	abbreviation
Correct	40.7%	42.6%	16.7%
Incorrect	—	58.3%	41.7%

nary. As for the company name identification results, correct and incorrect matches were analyzed according to the types of matches as shown in **Table 4**. To reduce the number of incorrect matches caused by company-name prefixes and suffixes, it is necessary to store more familiar company names in the MAJESTY dictionary, which enables stricter matching conditions.

As for TYPE-III parts of speech ambiguities presented in Fig. 6, about half were due to proper names that could be either a person name, place name, or company name. A total of 75% of parts of speech ambiguities caused by multiple kinds of proper names was resolved in the output of the proper name identification and grouping program. An example can be seen in the word "ジョーンズ" (see Fig. 4), whose part of speech was potentially either a family name (NOUN-FAMNAME) or a company name (NOUN-COMPANY), but which was successfully recognized as part of a company name by the proper name identification program. (Segments comprising the company name are grouped between <GRP-1> and </GRP-1> with a part of speech <GPAR>COMPANY</GPAR>.) Thus, the grouping result helped to disambiguate multiple parts of speech of proper names. Since the grouping algorithm had little to do with the way words were segmented, as long as prefixes and suffixes were correctly segmented, it also improved segmentation accuracy by putting together words that were incorrectly segmented by MAJESTY. By combining MAJESTY and the grouping results, the overall preprocessor accuracy reached 98.8%.

4.3 Processing speed

MAJESTY is written in C. It processes 732 characters per second on a SUN SPARCstation IPX, and takes an average of 1.4 seconds per newspaper article. The proper name identification and grouping program is written in JGAWK (Japanese GNU AWK). It processes 62 characters, or 7.8 grouped words per second.

5. Conclusions

The original morphological analyzer was modified to disambiguate its output and to locally pack ambiguous segments and parts of speech. The modified morphological analyzer, MAJESTY, performs with greater than 98% accuracy in the corporate joint ventures domain. Segments comprising company names, person names, place names, and numeric expressions were identified and grouped together. Company names have been identified with over 80% in both recall and precision. Person and place names have also been recognized with over 90% accuracy. MAJESTY, the grouping program, and the parser interface program comprise a Japanese preprocessor which has been successfully integrated into the SHOGUN and TEXTRACT systems for information extraction in the TIPSTER domains of corporate joint venture and microelectronics.

From the point of view of parsing, the benefits of introducing the Japanese preprocessor can be summarized as follows: (1) A parser can analyze Japanese texts in the same way it analyzes a word-based language such as English. (2) A parser can easily process as many ambiguities as necessary and select the correct segmented words, tagged with a correct part of speech, since ambiguities are packed locally and ranked in a likely order. (3) A parser can recognize a proper name as if it were a single word, since the proper name is already identified and grouped in the preprocessor.

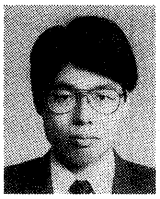
Acknowledgements The authors wish to express their appreciation to Jaime Carbonell who has given us the opportunity to pursue this research at the Center for Machine Translation, Carnegie Mellon University. We also thank Michael Mauldin and Eric Nyberg for many helpful suggestions.

References

1) Tomita, M., Kee, M., Mitamura, T. and Carbonell, J.: Linguistic and Domain Knowledge Sources for the Universal Parser Architecture,

Terminology and Knowledge Engineering, Czap, H. and Galinski, C. ed., pp. 191-204, INDEKS Verlag, Frankfurt (1987).
2) Tomita, M.: *Efficient Parsing for Natural Language*, Kluwer Academic Publishers (1986).
3) Kitani, T.: A Japanese Proofreading and Indexing System for Document Creating Systems (In Japanese), *The Journal of the Institute of Image Electronics Engineers of Japan*, Vol. 17, No. 5, pp. 337-345 (1988).
4) Kitani, T.: An OCR Post-processing Method for Handwritten Japanese Documents, *Natural Language Processing Pacific Rim Symposium*, pp. 38-45 (1991).
5) Nagao, M. ed.: *Nihongo Joho Shori* (In Japanese), 2nd ed., IECE, pp. 86-112 (1985).
6) Shudo, K., Narahara, T. and Yoshida, S.: A Structural Model of Bunsetsu for Machine Processing of Japanese (In Japanese), *IEICE Trans.*, Vol. J62-D, No. 12, pp. 872-879 (1979).
7) Yoshimura, K., Takeuchi, M., Tsuda, K. and Shudo, K.: Morphological Analysis of Japanese Sentences Containing Unknown Words (In Japanese), *Trans. IPS Japan*, Vol. 30, No. 3, pp. 294-300 (1989).
8) Matsumoto, Y., Kurohashi, S., Taegi, H. and Nagao, M.: *JUMAN Users' Manual version 0. 8* (In Japanese), Nagao Laboratory, Kyoto University (1992).
9) Kitani, T.: A Japanese Morphological Analyzer with a Proper Noun Detection Algorithm (In Japanese), *IPSJ SIG Reports*, Vol. 92, No. 55, pp. 73-80 (1992).
10) Rau, L. and Jacobs, P.: Extracting Company Names from Text, *Seventh IEEE Conference on Artificial Intelligence for Applications*, Vol. 1, pp. 29-32 (1991).
11) Kitani, T. and Mitamura, T.: A Japanese Preprocessor for Syntactic and Semantic Parsing, *Ninth IEEE Conference on Artificial Intelligence for Applications*, pp. 86-92 (1993).
12) Komachi, Y.: International Standardization Overview on Document Description Languages and Fonts (In Japanese), *Joho Shori*, Vol. 32, No. 10, pp. 1110-1117 (1991).

(Received January 5, 1993)
(Accepted November 11, 1993)



Tsuyoshi Kitani was born in Hiroshima, Japan on February 22, 1960. He received B.E. degree in Electrical Engineering from Keio University in 1983. He joined NTT Yokosuka Laboratories in 1983, then transferred to NTT Data Communications Systems Corp. in 1988. His research interests include natural language processing and expert systems. He has been a visiting researcher at the Center for Machine Translation, Carnegie Mellon University since September, 1991. He is a member of IPSJ.

Teruko Mitamura is a native of Japan, and received a Ph. D. in Linguistics in 1989 from University of Pittsburgh. She is currently a member of the research faculty at the Center for Machine Translation, Carnegie Mellon University (CMU), and has played a key role in several machine translation research projects since the foundation of the Center in 1986.