Building a Japanese Parsed Corpus while Improving the Parsing System

Sadao Kurohashi and **Makoto Nagao**
Department of Electronics and Communication, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, 606, Japan
{kuro, nagao}@kuee.kyoto-u.ac.jp

Abstract

In January 1996, we started a project to construct a Japanese parsed corpus and to simultaneously improve a morphological analyzer and a parser. In this project, human annotators are not only correcting the erroneous analyses produced by the parsing system, but also improving the parsing system/grammar: finding problematic fixed expressions, picking up phrases which have exceptional functions, and classifying unseen types of clauses, and so on.

1 Introduction

Corpus-based methods have become a central paradigm in the present-day NLP. Corpus-based NLP usually takes the following steps:

1. plain corpora are first assigned tags automatically (this paper limits the scope to part-of-speech tags and syntactic tags),
2. those tags are corrected by human annotators, making correctly tagged corpora,
3. tagged corpora are used to construct or improve NLP system/grammar.

Corpus construction projects so far have not fully exploited the automatic parser. In the case of Penn Treebank Project (Marcus et al., 1993), the employed parser, Fidditch (Hindle, 1989), was only expected to chunk the input into a string of trees, not to make a whole syntactic tree for the input. Annotators' task was then to glue together the syntactic chunks produced by the parser. In the case of ATR/Lancaster Treebank Project (Black et al., 1996), the employed parser produced the 'parse forest' for the input, and it was left to the annotators to choose the proper parse within the forest.

However, if a parser is available which can make a unique parse for the input accurately, the situation changes. One advantage in using such a powerful parser is the reduction of annotators' labour. Furthermore, there is another important advantage in using a powerful parser. If annotators correct the unique parses of the inputs, the process is almost equivalent to the detailed investigation of the performance of the parser. Accordingly, deficiencies of the system and/or grammar can be found simultaneously with the corpus construction process. That is, the process 2 and 3 above can be to some extent merged.

Apparently, the accuracy of the original parser is very important. If the parser produces pell-mell erroneous analyses often, it is hard to locate the essential problem of the parser, and also the manual correction process becomes much more troublesome. In the case of Japanese language processing, we have developed a morphological analyzer, JUMAN, and a parser, KNP (Kurohashi and Nagao, 1994), and have been improving them continuously. Consequently, their performance of producing unique parses for real world texts has been very close to the satisfactory level. Then, in January 1996 we started a project of constructing Japanese parsed corpus and improving the morphological analyzer and the parser simultaneously. In this paper, we describe the progress of the project up to date.

2 Overview of the Project

2.1 Information Assigned to the Corpus

The syntactic information in the corpus is based on dependency relations defined on *bunsetsu* unit,

* 0 1D						
沸点	ふってん	沸点	名詞	普通名詞	*	*
(*hutten* 'the boiling point'			Noun	Common Noun)		
が	が	が	助詞	格助詞	*	*
(*ga*			Postposition	Case Marker)		
* 1 4P						
高い	たかい	高い	形容詞	*	イ形容詞アウオ段	基本形
(*takai* 'high'			Adjective		I-adjective-type	Basic form)
、	、	、	特殊	読点	*	*
((comma)			Special Symbol	Comma)		
* 2 3D						
多く	おおく	多く	副詞	*	*	*
(*ooku* 'many'			Adverb)			
の	の	の	助詞	名詞接続助詞	*	*
(*no*			Postposition	Nominal-connector)		
⋮	⋮	⋮				

Figure 1: An example sentence with the morphological and syntactic information.

according to the parser employed, KNP. *Bunsetsu* is a commonly used linguistic unit in Japanese traditional grammar, consisting of one or more adjoining content words and zero or more following functional words.[1] Dependency grammar formalism has been used in KNP because of its suitability to Japanese language in the parsing process. However, it also has an important advantage in corpus contraction; lack of internal, artificial nodes, like NP, VP, makes annotators' task much easier.

The information assigned in the parsed corpus is as follows:

Morphological Information :

- boundaries between words,
- pronunciation, basic form, POS, conjugation type, conjugation form of each word.[2]

Syntactic Information :

- boundaries between bunsetsu's,
- governor bunsetsu of each bunsetsu, and one of the following three relations between the bunsetsu and the governor bunsetsu:

[1] The English equivalent for bunsetsu would be a small noun phrase, a prepositional phrase, and a verb phrase consisting of auxiliary verbs and a main verb, and so on.

[2] Japanese morphological grammar in JUMAN has 43 POS-set, 33 conjugation types, and about 15 conjugation forms for each conjugation type. For POS set, two level classification has been done. For example, "Noun" defined as the upper level class is classified into "Common Noun", "Proper Noun", "Adverbial Noun", and so on.

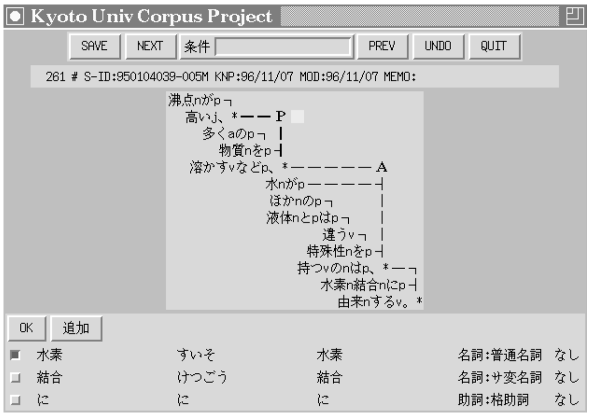


Figure 2: User interface for manual correction.

- **D** : predicate-argument/adjunct relation or head-modifier relation,
- **P** : coordination relation,
- **A** : appositive relation.

Figure 1 shows an example sentence with full information in the corpus (lines parenthesized are inserted for gloss, not included in the corpus). The original sentence is composed of words in the first column. Each line starting with a word shows its morphological information; each item in a line shows a word occurred in the original sentence, pronunciation, basic form, upper level POS, lower level POS, conjugation type, and conjugation form, respectively. A line starting with '*' shows a bunsetsu-boundary. The number next to '*' shows the ID of a bunsetsu consisting of the words between the line and the next '*' line; the

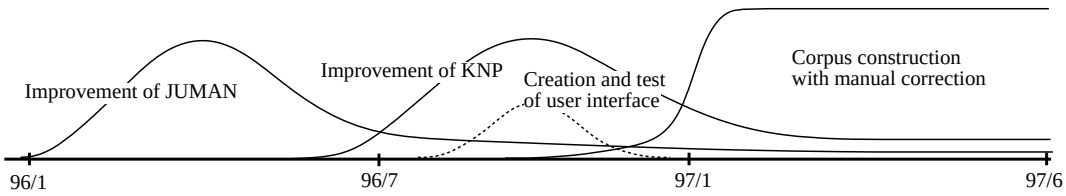


Figure 3: The schedule of the project.

next number and the character shows the ID of its governor bunsetsu and their type of relation.

These information are first assigned by JUMAN and KNP automatically, and then corrected by annotators using a mouse-based interface (Figure 2). The upper part of the interface window is for the dependency analyses correction; the lower part of the interface window is for the correction of the morphological analyses and the bunsetsu segmentation.

2.2 Progress to Date

The project started in January 1996 (Figure 3). In the first six months, we had intensively modified the morphological analyzer JUMAN (see Sec. 3).

Then, in the next six months, we modified the Japanese dependency structure analyzer KNP (see Section 4). In this period, we also made a mouse-based interface for manual correction, and did some trials of manual correction.

Then, we actually started manual correction process in January 1997. As of June 1997, we constructed about 10,000 sentences of parsed corpus. Currently two annotators are attending to the project, both of them received their master's degree in Japanese linguistics. Their current task is to correct the analyses of texts in 80% of their work time, and to investigate the reasons of erroneous analyses in the remaining time. An average speed of their correction is approximately 40 sentences ($\simeq$ 1,000 words) per hour. We have a weekly meeting to discuss the problems of the current system/grammar and modify them periodically.

The final goal of the project is to construct 200 thousand sentences ($\simeq$ 5 million words) of parsed corpus. As an original text of the corpus, we are currently using newspaper articles (Mainichi Newspaper, 1995). However, since the sentential style in newspaper articles is biased, we plan to use other types of text from now on: fiction, essay, scientific documents, and so on.

3 Morphological Analyzer JUMAN

A robust morphological analysis is needed for Japanese language since Japanese sentences have no explicit separators between words and the words such as verbs and adjectives have complex conjugation. The Japanese morphological analyzer JUMAN was developed in 1992, and it has been used by over one hundred researchers and groups internationally.

At the first stage of the project, we intensively modified JUMAN, making it accurate enough as a basic tool for corpus construction.

3.1 Basic Algorithm of JUMAN

Japanese morphological analysis is usually done using bigram information, which is often referred to as connectivity of pairs of words. JUMAN first recognizes all the possible character combinations which constitute a word in the given input by looking up the word dictionary, and then finds possible connections between adjacent words by checking the connectivity dictionary.

In many cases, however, there are many possible word segmentation patterns for the input which meet the connectivity constraints. To handle this problem, costs are assigned to each word (or class of words) and each connectivity between words in JUMAN. The higher the cost is given, the lower the word frequency, or the frequency of the two words in an adjacent position. By summing up those costs, JUMAN determines the most plausible answer (the most plausible word string) for the input.

3.2 Handling of Fixed Expressions

The erroneous analyses of the above algorithm are mainly due to the cost calculation, that is, when the correct answer is not assigned the minimal cost. In the conventional version of JUMAN, the only way of resolving such erroneous analyses was to adjust costs of some specific words and/or con-

nectivities. However, such a fine adjustment often causes side effects; a cost-adjustment for an erroneous analysis might cause other erroneous analyses to other inputs. For this reason, the total performance of the conventional version of JUMAN could not be improved any more by such cost adjustments.

The erroneous analyses by cost calculation often occur for a long string of *kana* characters (Japanese alphabet). This is because a small set of kana characters are used much more frequently than *kanji* characters (Chinese characters) in Japanese lexicon. However, strings of kana characters usually compose some fixed expressions consisting of two or more words, and in most cases their segmentations are not ambiguous when they are seen as a whole. Accordingly, if the system can handle a word string as a whole, defining its total cost and its connectivity, most of the erroneous analyses by cost calculation can be resolved without side effects.

The main modification to JUMAN we did in the project was the enhancement of the system to handle a word string as a whole, and to find and enter problematic fixed expressions which were analyzed incorrectly by the normal cost calculation. The current dictionary in JUMAN contains about 120 fixed expressions. Some examples are as follows:

- *tameshi* 'instance' *ga* 'case marker' *nai* 'there is no'
- *ka* 'if' *dou* 'how' *ka* 'if'
- *ni* 'case marker' *mo* 'also' *sukoshi* 'some'

For example, if the word string "*tameshi ga nai*" has not been registered in the dictionary, "*tameshiganai*" is incorrectly segmented into "*tame* 'for' *shiganai* 'poor' " by the normal cost calculation.

The registration of such problematic expressions is currently continued in the corpus construction process.

3.3 Current Accuracy of JUMAN

To see the current accuracy of JUMAN, the automatic analyses by JUMAN and the manually corrected analyses of about 10,000 sentences were compared. The counting is based on the words in the manually corrected analyses, and the upper level of POS classifications only are compared. By this criteria, the current accuracy of JUMAN is around 99.0%.

4 Dependency Structure Analyzer KNP

Simultaneously with the construction of a Japanese parsed corpus, the goal of the project is to make the Japanese syntactic analyzer KNP more accurate. In the following, several issues in KNP are discussed, including its recent improvements.

4.1 Basic Grammar Formalism of KNP

KNP is based on a dependency grammar defined on bunsetsu unit. The advantage of considering bunsetsu as a linguistic unit of a grammar is as follows:

- Syntactic behavior of bunsetsu is much clearer (more specified) than that of word so that the grammar construction based on bunsetsu unit is easier than based on word unit.
- Behavior of some bunsetsu cannot be explained compositionally by its components (words). For example, "*kaku* 'write' *mono* 'thing' *no* 'of' " means 'though writing'. In such cases, it is natural to define syntactic function to a bunsetsu, not to a word.

KNP first converts a word string segmented by JUMAN into a bunsetsu string. This process can be done by checking POS's of words without ambiguity in most cases.

Japanese nominal arguments have heavy characteristics: the order flexibility and the omissibility. The dependency grammar formalism can cope with these characteristics. This is because the dependency relation between a predicate and an argument remains as it is, even if arguments are scrambled or some of them are omitted.

Japanese language is head-final, that is, a bunsetsu depends on another bunsetsu to its right (not necessarily the adjacent bunsetsu). Accordingly, syntactic ambiguity in dependency grammar formalism arises when a bunsetsu can depend on two or more bunsetsu's which follow in a sentence. KNP resolves such ambiguity by using several heuristic rules described in the following sections.

4.2 Treatment of Exceptional Bunsetsu's

In some contexts, certain bunsetsu's do not work as defined by pure syntax. For example

([···] establishes a context):

- *dekiru* 'can do' *dake* 'only' [*shizukani* 'quietly']
  — works as an adverb like 'as ··· as possible'.
- *utsukushiku* 'beautiful' [*saku* 'bloom']
  — works as an adverb like 'beautifully'.
- [··· *wo* 'case marker'] *henkan* 'transformation' , — works as an verb like 'transform'.
- [··· *wo* 'case marker'] *megutte* 'tour'
  — works as a postposition like 'as to'.

When a bunsetsu works exceptionally like in the above examples, it can be usually recognized by checking the surrounding bunsetsu's. Therefore, we enhanced KNP to have a regular-expression pattern matching facility so that several features marking exceptionality can be given to a bunsetsu according to its local context. A regular-expression pattern consists of:

- a pattern for the bunsetsu to which some feature is given (a pattern for a bunsetsu specifies the word sequence in the bunsetsu),
- patterns for its preceding bunsetsu sequence,
- patterns for its following bunsetsu sequence.

For example, if a bunsetsu consists of two words, *dekiru* 'can do' and *dake* 'only', and the next adjacent bunsetsu consists of a verb or an adjective, the bunsetsu is given a *non-predicate feature*. A bunsetsu with this feature is not treated as a predicate though the POS of its content word is verb, so that it does not govern nominal arguments.

There must be many other expression like the above examples, and it is important in the step-by-step improvement of the parser to find such expressions and treat them appropriately in the grammar. Such process can be done along with the manual correction of syntactic analyses in the corpus construction.

4.3 Treatment of Coordinate Structures

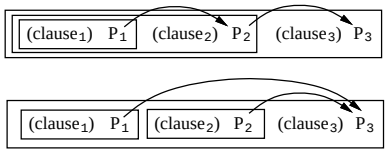
Coordinate structures (CSs) cause a serious problem to the syntactic analysis. This is because not only do CSs themselves have complex scope ambiguity, but also sentences with CSs tend to be long and the combination of scope ambiguities and the intrinsic syntactic ambiguities in long sentences can easily cause combinatorial explosion.

To solve this problem, Kurohashi and Nagao have developed a unique, efficient method of resolving CS scope ambiguities (Kurohashi and Nagao, 1994). The underlying assumption of this method is that conjoined phrases/clauses/sentences exhibit parallelism, that is, they have a certain similarity in the sequence of words and their grammatical structures as a whole. Based on this assumption, they devised a dynamic programming algorithm which determines conjuncts by finding the two most similar word-sequences on the left and the right of a conjunction.

This method can detect CSs' scopes accurately, which leads to the substantial reduction of the syntactic ambiguity in a sentence. Therefore, KNP analyzes CSs in the input first by using this method, and then it detects the dependency structure of the input keeping the detected CSs' scopes.

4.4 Treatment of Subordinate Clauses

Another serious problem in parsing sentences is the treatment of subordinate clauses. When there are three or more clauses in a sentence, we have to detect their correct scopes. In dependency grammar formalism, scopes of clauses can be expressed by dependency relations among the predicates of the clauses as follows (note that a predicate is final in a clause):



To this problem, Minami proposed a very effective preference rule (Minami, 1993). He classified several types of clauses into three classes depending on their strength, e.g., *-keredo* 'although ···' is classified into the strongest class, *-tsutsu* 'while ···' is classified into the weakest class. Based on his clause classification, he claimed that a weaker clause cannot contain a stronger clause in its scope, that is, the predicate in a stronger clause does not depend on the predicate in a weaker clause.

This claim fits the behavior of Japanese clauses very well, i.e., it works almost as a constraint. The problem with adopting Minami's study to the real parsing is that he just listed representative types of clauses to each class. In the project, therefore, we extended Minami's clause classification to be

a good coverage on real world texts, and incorporated the preference rule based on the clause classification into KNP. The update of the clause classification is currently continued in the corpus construction process.

4.5 The Remaining Problem

KNP can resolve ambiguities in coordinate structures and subordinate clauses as described in the preceding sections. Then, the major remaining ambiguity is the predicate-argument relation in embedded sentences. This ambiguity occurs, for example, when the input is "N_1 V_1 N_2 V_2", and N_1 can depend either on V_1 or V_2, satisfying the dependency constraint.

This type of ambiguity can be resolved by using the case-frame information, which specifies what types of nouns can fill each case slot of a predicate. However, there is no Japanese case-frame dictionary of wide coverage so far. Accordingly, KNP currently uses the naive preference rule which treats a bunsetsu that can depend on two or more bunsetsu's as if it depended on the nearest possible one only. (This rule is almost the same as *right association* principle in parsing English sentences.)

However, even if a case-frame information is not available, it might be possible to use corpus-based technique, that is, to use the co-occurrence frequencies of N_1 and V_1, and N_1 and V_2 in corpora as a rating cue (exhaustive study of such technique has been done for English PP attachment problem). When some amount of manually corrected texts is accumulated in the project, such a technique is promising to this type of ambiguity as well.

4.6 Current Accuracy of KNP

To see the current accuracy of KNP, the automatic analyses by KNP and the manually corrected analyses of about 10,000 sentences were compared. The counting was based on the pair of bunsetsu's which have a dependency relation in the manually corrected analyses. To evaluate the native performance of KNP, if the morphological analysis of both or either bunsetsu in a pair is incorrect, the pair is removed from the counting. Furthermore, we removed the pair of the last bunsetsu and the second last bunsetsu in each sentence, since this pair is always analyzed correctly based on the head-final feature. As for the remaining pairs of bunsetsu's in the manually corrected analyses, if the same dependency relation exists in the automatically analyzed corpora, it is counted as a correct analysis; if not, it is counted as an incorrect analysis. By this criteria, the current accuracy of KNP is in the range from 87 to 90%.

5 Conclusion

This paper has described the ongoing corpus project at Kyoto University, i.e., the construction of Japanese parsed corpus and simultaneous improvement of the morphological analyzer JUMAN and the parser KNP.

In October 1996, we released the enhanced version of JUMAN, JUMAN3.0. In May 1997, we released the enhanced version of KNP, KNP2.0, and we also opened about 10,000 sentences of parsed corpus. All these resources are available from `http://www-nagao.kuee.kyoto-u.ac.jp/`.

Acknowledgments

The project was partially supported by Grant-in-Aid for Scientific Research 07558046 and JSPS-RFTF96P00502 (The Japan Society for the Promotion of Science, Research for the Future Program)

References

Ezra Black, et al. 1996. Beyond skeleton parsing: producing a comprehensive large-scale general-English treebank with full grammatical analysis, In *Proceedings of COLING*, pages 107–112, Copenhagen, Denmark.

Donald Hindle. 1989. Acquiring disambiguation rules from text. In *27th ACL*, pages 118–125, Vancouver, British Columbia, Canada. Association for Computational Linguistics, Morristown, New Jersey.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4) pages 507–534.

Mitchell P. Marcus, et al. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2), pages 313–330.

Fujio Minami. 1993. A grammar of contemporary Japanese. Taishukan Publishing, Tokyo.