Building a Large-Scale Japanese CFG for Syntactic Parsing

Tomoya Noro **Taiichi Hashimoto** **Takenobu Tokunaga** **Hozumi Tanaka**
Department of Computer Science,
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, Tokyo
{noro,taiichi,take,tanaka}@cl.cs.titech.ac.jp

Abstract

Large-scale grammars are a prerequisite for parsing a great variety of sentences, but it is difficult to build such grammars by hand. Yet, it is possible to derive a context-free grammar(CFG) automatically from an existing large-scale, syntactically annotated corpus. While being seemingly a simple task at first sight, CFGs derived in such a fashion have hardly ever been applied to an existing systems. This is probably due to the great number of possible outputs, i.e. parse results(high ambiguity). In this paper, we analyze some causes of this high ambiguity, and we propose a policy for building a large-scale Japanese CFG for syntactic parsing, capable to decrease ambiguity. We end the paper with an experimental evaluation of the obtained CFG.

1 Introduction

Large-scale grammars are a prerequisite for parsing a great variety of sentences, but it is difficult to build such grammars by hand. Yet, it is possible to build a context-free grammar(CFG) by deriving it from a syntactically annotated corpus. Many such corpora have been built recently to obtain statistical information concerning corpus-based NLP technologies. For English, it is well known that a CFG derived from the Penn Treebank corpus(Marcus et al., 1993) can parse sentences(especially long sentences) better than any of the hand-crafted grammars(Charniak, 1996). Actually, there have been quite a few studies concerning this kind of grammar. In the case of Japanese, however, there is no large-scale syntactically annotated corpus like the Penn Treebank corpus. Although Shirai et al. proposed a method to derive a CFG from the EDR corpus [1], guessing nonterminal symbols to be assigned automatically to each intermediate node (Shirai et al., 1995). It still remains to be seen how accurately these nonterminal symbols can be guessed. We need a large-scale, syntactically annotated corpus of Japanese in order to derive a large-scale Japanese CFG akin to (Charniak, 1996).

However, even if a large-scale, syntactically annotated corpus were already available, a CFG derived from it poses a problem, in as it creates a great number of possible readings (in average more than 10^{12}, according to our preliminary experiment). Too many results reduce, of course, the parsing accuracy. They also decrease the speed of parsing, and they require more memory to parse and to store long sentences. Although Charniak has removed the CFG rules, occurring only once in the corpus to avoid such problems, this is not enough, as the rules that occur more than once may also increase ambiguity.

Since the sentences of a normal, syntactically annotated corpus have “semantically correct” structures, the derived grammar creates many parse results, representing the different possible reading, i.e. meanings. A syntactic parser does not deal with se-

[1] the EDR corpus(EDR, 1994) being a bracketed corpus, with only skeletal structures for each sentence. The intermediate nodes of the tree are not annotated with nonterminal symbols.

mantics. Hence, it is difficult to deal with ambiguities of that sort. On the other hand, if the parser creates many different structures, it becomes difficult to disambiguate the results, even if semantic analysis is carried out after the syntactic parsing.

We assume Syntax-Driven Semantic Analysis(SDSA)(Jurafsky and Martin, 2000), that is, syntactic analysis, based on a large-scale CFG, is followed by a semantic analysis. Since the parse results are sent to a subsequent SDSA phase, the number of parse results should be as small as possible. Therefore, it is necessary to build a CFG that minimizes the results during syntactic parsing. We attempt to build such a CFG from a syntactically annotated corpus, by using the following method:

1. Derive a CFG from an existing, syntactically annotated corpus.
2. Analyze the causes of increased ambiguity.
3. Create a policy for modifying the corpus.
4. Modify the corpus according to the policy, and derive again a CFG from it.
5. Repeat steps (2), (3) and (4) until all problems are solved.

While repeating the steps (2) - (4) is labor-intensive, it is very important to do so in order to build an adequate, large-scale CFG for syntactic parsing.

In this paper, we propose a method to build such a large-scale Japanese CFG, capable to minimize parse results, assuming furthermore, that the parse results will be sent subsequently to the SDSA component. Note that the CFG described in this paper does not perform any semantic analysis, it deals with syntax only. While our syntactic structures might look a bit odd from a semantic point of view, they are useful for keeping ambiguity low during syntactic parsing.

2 Causes of Increased Ambiguity

A great number of results are produced during the syntactic parsing of the CFGs, derived from syntactically annotated corpora. To decrease these ambiguities (i.e. the number of parse results), we start by analyzing their causes. There are four main reasons for this situation:

Human Errors: Human annotators sometimes make mistakes when annotating the syntactic structure of a sentence. If there are mistakes in the corpus, a CFG derived from it, is likely to produce an incorrect structure.

Inconsistency: There may be some contradictions concerning the structures. Since large-scale corpora are usually built incrementally and by several annotators, it takes a long time to build them, and the annotated structures may vary from annotator to annotator, unless the annotation policy is clearly defined. Hence, a CFG derived from an inconsistent corpus can yield many parse results with inconsistent structures.

Lack of Syntactic Information: Lack of syntactic information may yield incorrect syntactic parses. Since CFG rules represent only structures of subtrees of depth one(relation between a parent node and some child node), no other syntactic information is used during syntactic parsing. Yet, in the case of Japanese, a verb phrase can be an adnominal phrase, a continuous, or subordinate clause. In order to decide which one to choose, one has to consider the verb conjugation or the particles (postpositions) at the end of the phrase. In a sentence like "*boushi wo kabutteiru hito wo mita* (I saw the person wearing a hat)", the verb phrase "*boushi wo kabutteiru*" could be an adnominal phrase, because the conjugation of the verb "*kabutteiru*" is an adnominal form. If no information concerning verb conjugation can be assigned at the intermediate nodes of the subtree covering the verb phrase, it is not clear whether the verb phrase is an adnominal phrase or a continuous clause. Syntactic information outside of a CFG rule should be assigned to each node if necessary.

Need for Semantic Information: Semantic information is necessary for disambiguation in some cases(e.g. PP attachment problem). In the case of a phrase like "*kare no me no iro*", one cannot decide whether the adnominal phrase "*kare no*" should be attached to the noun "*me*"(the clause meaning "color of his eyes"), or to the noun "*iro*"(the clause meaning "his color of eyes") by relying solely on syntactic information.

In order to build a large-scale CFG that creates a minimal number of parse results, such issues have to be addressed. Since the causes of (1) and (2) are annotation errors, they need to be corrected manually. On the other hand, since (3) and (4) are not errors, they can be handled by modifying the structures in the syntactically annotated corpus and by deriving the CFG from this newly-annotated corpus. We propose a policy for modifying syntactically annotated corpora in the following section.

3 Policy for Modifying the Corpus and the Grammar

3.1 Definitions

First, we define three types of ambiguities by modifying slightly the definition proposed by (Komagata, 1997):

1. Multiple syntactic structures of the sentence with distinct meanings. The fourth cause of the increased ambiguity described in the previous section belongs to this type.

2. Multiple syntactic structures of the sentence that are semantically equivalent, or syntactically incorrect structures due to insufficiencies of the CFG. The first, second, and third cause of the increased ambiguity is of this kind.

3. One syntactic structure of the sentence with multiple meanings(semantic interpretation).

In this paper, we refer to these ambiguities as TYPE-1, TYPE-2, and TYPE-3, respectively.

TYPE-2 is spurious ambiguity, which can only be removed by modifying the CFG in order to avoid such an ambiguity. Furthermore, since it is difficult to eliminate the ambiguity of TYPE-1 without recourse to semantic information during syntactic parsing, we build a large-scale CFG, leaving the disambiguation to the subsequent SDSA phase, by converting TYPE-1 ambiguity into TYPE-3 whenever possible(Church and Patil, 1982; Nitta et al., 1984; Jensen and Binot, 1987). Since some semantic interpretations can be grouped if TYPE-1 ambiguity is converted into TYPE-3, the number of parse results decreases. This improves the speed of parsing and reduces the use of memory. In addition, the possible semantic interpretations the parse result includes are easily extracted if they are grouped together under certain policy. This being so, it becomes obvious what kind of ambiguity has to be disambiguated in the subsequent SDSA phase.

In the next section, we discuss the basic policy for modifying the CFG and the syntactically annotated corpus.

3.2 Modification Policy

The major ambiguities of TYPE-1 are the following:

1. Compound noun structure

2. Adverbial phrase attachment

3. Adnominal phrase attachment

4. Conjunctive structure

In this section, we discuss whether each one of these ambiguities could and should be converted into TYPE-3 or not.

3.2.1 Compound noun structure

In general, it is difficult to disambiguate the structures of compound nouns without any semantic information. Shirai et al. modify their CFG to produce a right linear binary branching tree for a compound noun during the parse(Shirai et al., 1995) [2]. We modify the structures of the compound noun in the same way, thus converting the ambiguity under study into a TYPE-3 ambiguity.

3.2.2 Adverbial and Adnominal Phrase Attachment

We consider annotating the syntactic structure of a sentence like "*oubei shokoku ha nihon no ryuutsuu seido no kaizen wo motometeiru*(The European community asks Japan to improve its systems of distribution)" to discuss the adverbial phrase- and the adnominal phrase attachment. A bottom-up style parsing works as follows:

[2] Instead of the term "compound noun", Shirai et al. use the term "compound word", meaning by that term any constituent covering an identical part-of-speech sequence (e.g. a noun sequence). Our term "compound noun" refers to the fact that the constituent under study acts as a noun and consists of nouns, suffixes, prefixes, etc.(there is no need for an identical POS sequence).

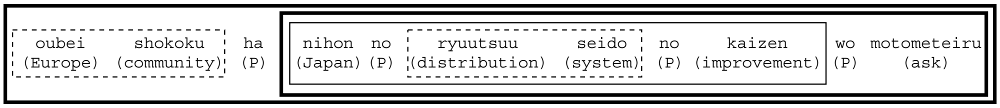

Figure 1: Syntactic structure of the sentence "*oubei shokoku ha nihon no ryuutsuu seido no kaizen wo motometeiru*(The European community asks Japan to improve its system of distribution)"

1. Two compound nouns "*oubei shokoku*" and "*ryuutsuu seido*" are made (the dashed rectangles in Figure 1).

2. Two adnominal phrases "*nihon no*" and "*nihon no ryuutsuu seido no*" are attached to the nouns "*ryuutsuu seido*" and "*kaizen*", respectively (the thin solid rectangle in Figure 1).

3. Two adverbial phrases "*oubei shokoku ha*" and "*nihon no ryuutsuu seido no kaizen wo*" are attached to the verb "*motometeiru*" (the thick solid rectangles in Figure 1).

Thus, the compound noun is the lowest level, while the adverbial phrase attachment is the highest level. Adnominal phrase attachment is somewhere in between. Since we have decided to convert the ambiguity of compound noun structure(at the lowest level) to TYPE-3, we convert the ambiguity concerning the adnominal phrase attachment(in the intermediate level) to TYPE-3, while the ambiguity concerning adverbial phrase attachment(at the highest level) remains a TYPE-1 ambiguity.

There are basically two types of ambiguity for adnominal phrase attachment: one changes the range of the adverbial phrases, while the other does not. In the case of a sentence like "*atarashii kankyou he no tekiounouryoku wo shiraberu*(I investigate the possibility to adapt to a new environment)", the range of the adverbial phrase attached to the verb "*shiraberu*" is the same, regardless of whether the adnominal phrase "*atarashii*" attaches to the noun "*kankyou*" or to the noun "tekiounouryoku". On the other hand, in the case of a sentence like "*10nen no rekishi wo motsu matsuri*(the festival with 10 years of history)", the range of the adverbial phrase attached to the verb "*motsu*" is different, depending on whether the the adnominal phrase "*10nen no*" attaches to "*rekishi*" or whether it attaches to "*matsuri*". Although we have proposed to handle the adnominal phrase attachment as TYPE-3 ambiguity, we will refrain from it when the range of the adverbial phrase changes(i.e. in case of "*10nen no rekishi wo motsu matsuri*").

To summarize, our policy for handling the ambiguity of adverbial and adnominal phrase attachment is as follows:

1. The ambiguity of adverbial phrase attachment is left as TYPE-1 ambiguity.

2. The ambiguity of adnominal phrase attachment is converted to TYPE-3 ambiguity when the range of adverbial phrase does not change, otherwise it is left as TYPE-1 ambiguity.

Since we believe that a different algorithm should be used to disambiguate adverbial phrase attachment and adnominal phrase attachment in Japanese, we have decided to deal with them separately. This means that the ambiguity concerning whether a phrase is an adverbial phrase or adnominal phrase remains during syntactic parsing. However, this increase of ambiguity is not very big. Actually, in Japanese it is relatively easy to discriminate between an adverbial and an adnominal phrase[3].

In the case of a sentence like "*douro no ryougawa ni ha mizu wo nagasu tame no mizo ga hottearimasu*(Drainage trenches are dug on both sides of the street)", our CFG can create four different structures as shown in Figure 2 (the structure (a) is the appropriate parse according to our policy). The dashed and solid rectangles denote verbs and adverbial phrases respectively, and the phrases are attached in the direction of the arrow. After syntactic parsing, adverbial phrase attachment(TYPE-1 ambiguity) will be disambiguated by selecting the seman-

[3] There are cases where this discrimination is not so easy though. For example, the adverb '*hobo*' can be an adverbial phrase in the case of a sentence like "*hobo owatta*(It has almost been finished)" while it can be an adnominal phrase in the case of a sentence like "*hobo zen'in ga kita*(Nearly everyone has come)", however, these cases are quite limited in number.

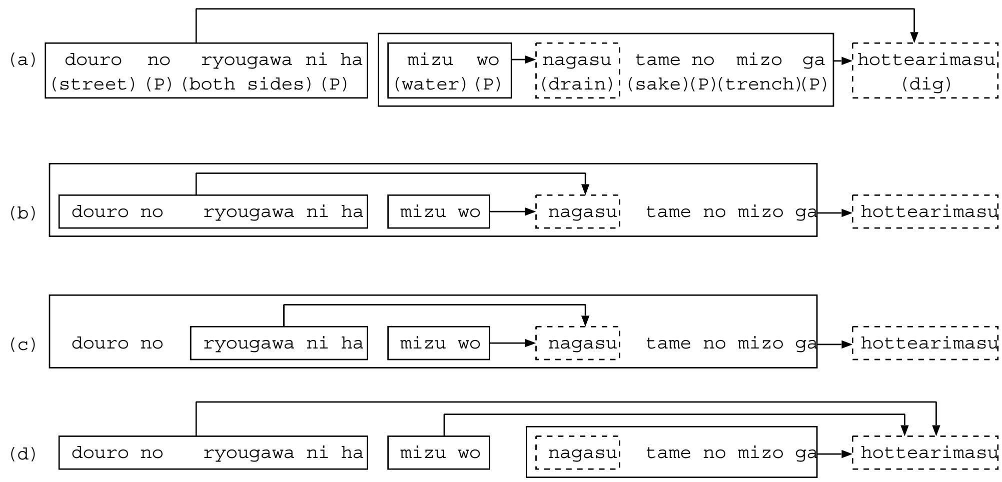

Figure 2: Syntactic structures of the sentence "*douro no ryougawa ni ha mizu wo nagasu tame no mizo ga hottearimasu*(Drainage trenches are dug on both sides of the street)" that our CFG can create.

tically correct parse among the possible ones. On the other hand, adnominal phrase attachment(TYPE-3 ambiguity) will be disambiguated by recreating the structure.

3.2.3 Conjunctive Structure

In general, parsing accuracy of the sentences containing conjunctive structures is significantly worse than that of sentences without such structures. Our preliminary experiment shows that the sentence accuracy of such sentences is only about half of the rest [4]. Coping with conjunctive structures is important for improving overall accuracy.

Since semantic information is necessary for analysis of conjunctive structures, it is difficult to disambiguate these structures in syntactic parsing. Kurohashi et al. propose a method that first detects conjunctive structures in a sentence, then analyzes the dependency structure of the sentence in order to disambiguate them(Kurohashi and Nagao, 1994). Contrary to their method, our CFG does not specify conjunctive structures before syntactic parsing, as they are assumed to be analyzed during the subsequent phase. Conjunctive structures are TYPE-3 ambiguities in our CFG.

According to (Kurohashi and Nagao, 1994) there are three types of conjunctive structures in Japanese:

1. conjunctive noun phrases
2. conjunctive predicate clauses
3. conjunctive postpositional phrases [5]

We discuss how to handle each one of them in the remainder of this section.

Conjunctive Noun Phrases

In case of a phrase like "*nihon to chuugoku no kankei* (the relationship between Japan and China)", two parse results are created: one states that the nouns "*nihon*(Japan)" and "*chuugoku*(China)" constitute pre- and post-conjuncts, while the other considers the nouns "*nihon*" and "*kankei*(relationship)" as conjuncts. It is difficult to disambiguate such sentence solely on the basis of syntactic information. Assuming that the pre-conjuncts("*nihon to*" in our example) are identified as adnominal phrases, we treat the structure of such conjunctive noun phrase as TYPE-3 ambiguity.

Conjunctive Predicate Clauses

The parsing accuracy of the sentences containing conjunctive predicate clauses are much lower than that of the sentences containing other types of conjunctive structures. This is so because it is difficult to determine whether two clauses are pre- or post-conjuncts or not. In the case of a sentence like "*uta wo utai, odori wo odoru*(I sing a song

[4]The definition of the accuracy of the sentences are described later.

[5]Kurohashi et al. refer to them as "incomplete conjunctive structures".

Table 1: The number of rules in the CFGs and the number of parse results

	# CFG rules	# non-terminal symbols	# terminal symbols	# parse results(avg.)
pre-modified	1,694	249	600	1.868×10^{12}
post-modified	1,949	279	600	9.355×10^{5}

and dance a dance)", some annotators consider it as a conjunctive structure while others claim that it is not, because the definition of conjunctive predicate clause is not clear. Assuming that pre-conjuncts are identified with adverbial phrases, determining whether two predicate clauses constitute pre- and post-conjuncts or not is left to the subsequent phase.

Conjunctive Postpositional Phrases

Since the particles and postpositions in a conjunctive postpositional phrase are often the same as the corresponding phrase, it does not seem too difficult to determine the appropriate structure. However, the parsing accuracy of the sentences containing conjunctive postpositional phrases is as low as that of the sentences containing conjunctive noun phrases in our preliminary experiment. This is so because the two phrases having the same particles or postpositions do not always constitute conjuncts. In the case of "*1ji ni eki ni iku*(I will go to the station at one o'clock)", although the two postpositional phrases "*1ji ni*(at one o'clock)" and "*eki ni*(to the station)" have the same particle "*ni*", they do not constitute conjuncts. Semantic information is necessary for determination of the appropriate structures. Assuming that the phrases are identified as adverbial phrases, determining whether the two phrases constitute conjuncts or not is left to the subsequent phase.

4 Evaluation

To evaluate the efficiency of the CFG modified according to our policy, we consider two aspects, both of which are important: the number of parse results derived by the CFG, and the accuracy of the parsing achieved by using our CFG. As mentioned earlier, it is important to decrease the number of parse results, as this speeds up the process while reducing memory load. It goes without saying that it is more important to increase the accuracy of the parsing rather than to speed up the process.

First, we picked up 8,911 sentences(on average about 20 morphemes in a sentence) from the EDR corpus(EDR, 1994) and manually annotated "semantically correct" structures in each sentences. Then we modified the structures according to the policy described above by an annotation tool(Okazaki et al., 2001). Two CFGs are derived from the pre- and post-modified corpus, and used to parse the part-of-speech(POS) sequences of the sentences by the MSLR parser(Shirai et al., 2000)[6]. We refer to the two CFG as pre- and post-modified CFG respectively. The number of rules in two CFGs and the number of parse results are shown in Table 1. The number of parse results considerably decreased(by 10^7 order) while the number of the CFG rules slightly increased[7].

Next, we ranked parse results by training the parser according to the probabilistic generalized LR(PGLR) model(Inui et al., 2000) using the 10-fold cross-validation: we train the parser using 8,020 sentences and evaluate it on the rest of the data[8]. Figure 3 shows the sentence accuracy(SA), which is defined as follows:

$$\text{SA} = \frac{\text{\# sentences parsed correctly}}{\text{\# sentences parsed}}$$

"Sentences parsed correctly" are the sentences in which all constituents are labeled correctly among the top-n parsing results(n is the value of the x-axis of Figure 3). Since the parse results are re-analyzed using the semantic information in the subsequent phase, the structure of the parse result must match the correct structure exactly. That is why we use this evaluation metric rather than the labelled precision and the labelled recall, which are commonly used in evaluation of parsing.

Assuming that the top-100 parse results are re-analyzed in the subsequent phase, the accuracy is

[6]Although the MSLR parser integrates morphological and syntactic analysis of unsegmented sentences, it can perform only syntactic parsing by giving POS sequences as inputs.

[7]The number of terminal symbols does not change because we have not modified any POS tags under our policy

[8]The CFGs are derived from all sentences(also from test data).

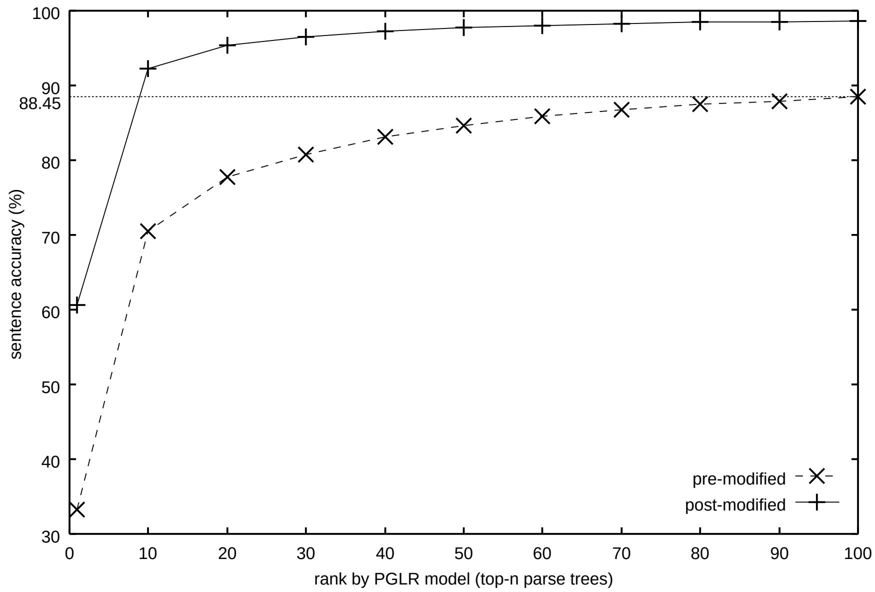

Figure 3: Sentence accuracy

98.6% using the post-modified CFG while the accuracy is only 88.5% using the pre-modified CFG, which means more than 10% of the sentences cannot be re-analyzed correctly when the pre-modified CFG is used for syntactic parsing. On the other hand, only the top-10 parse results are enough for the post-modified CFG to overcome the accuracy among top-100 parsing results using the pre-modified CFG. This result shows that the post-modified CFG is more useful for syntactic parsing than the pre-modified CFG and that our policy is also useful in building a large-scale Japanese CFG.

Additionally, we examined the top parse results of 100 randomly selected sentences from the post-modified corpus [9]. 96 sentences of these are correctly segmented into Japanese phrasal units(*bunsetsu*), and the dependency accuracy(the percentage of correct dependency units out of all dependency relations) is 89.23% [10], which rivals the other state-of-the-art systems using support vector machine, maximum entropy, etc(Kudo and Matsumoto, 2002; Kanayama et al., 2000; Uchimoto et al., 2000) [11] although we have not incorporated any semantic information yet [12]. We think that the accuracy will increase as soon as semantic information is incorporated in the subsequent phase. The method to incorporate semantic information is left for future research.

5 Conclusion

Although a large-scale grammar can be derived from a syntactically annotated corpus, in general, such grammars create a large number of parse results. The principal reason is that such grammars are not modified to sufficiently limit the ambiguity. We showed that a practical large-scale grammar for syntactic parsing can be built by investigating the causes of increased ambiguity and modifying grammars and corpora to remove the causes of such ambiguity.

In the future, we intend to look into the following:

1. Since it is not sufficient to modify the corpus and the grammar, we have to consider other

[9] In this experiment, the 100 randomly selected sentences are used for evaluation and the rest of the sentences are used for training the parser. A CFG is derived from all sentences.

[10] We assume every adnominal phrase attaches to the next noun if it is ambiguous which noun is attached to by the adnominal phrase.

[11] We cannot compare their model with ours with absolute equity because they use different corpus and carry out their experiment under different conditions

[12] Although the dependency accuracy using the pre-modified corpus is nearly the same, the semantic interpretation will fail in the subsequent SDSA phase, because many intermediate nodes are simply wrong.

problems. For example, particles and postpositions are often omitted in Japanese, causing problem in our CFG.

2. Since we assume that the parse results created by our CFG are re-analyzed in a subsequent SDSA phase, we have to provide a method for re-analysis of the parse results.

3. We have not considered the ambiguity at the morphological level. We have to review the word segmentation and the POS assignment to decrease the ambiguity at the syntactic level as well as at the morphological level. We are considering a POS system based on ChaSen, the well-known Japanese morphological analyzer(Matsumoto et al., 1997).

4. If we change the policy for annotating corpora and building grammars, we have to incorporate the changes into existing data. We are planning to construct a large support system for annotating corpora including database systems, annotation tools, etc.

References

Eugene Charniak. 1996. Tree-bank grammars. In *the 13th National Conference on Artificial Intelligence*, pages 1031–1036.

Kenneth Church and Ramesh Patil. 1982. Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, 8(3–4):139–149.

EDR, 1994. *EDR Electronic Dictionary User's Manual*, 2.1 edition. In Japanese.

Kentaro Inui, Virach Sornlertamvanich, Hozumi Tanaka, and Takenobu Tokunaga. 2000. Probabilistic GLR parsing. In Harry Bunt and Anton Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*, pages 85–104. Kluwer Academic Publishers.

Karen Jensen and Jean-Louis Binot. 1987. Disambiguating prepositional phrase attachments by using on-line dictionary definitions. *Computational Linguistics*, 13(3–4):251–260.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing*. Prentice-Hall.

Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun'ichi Tsujii. 2000. A hybrid Japanese parser with hand-crafted grammar and statistics. In *COLING 2000*, pages 411–417.

Nobo Komagata. 1997. Efficient parsing for CCGs with generalized type-raised categories. In *IWPT 97*, pages 135–146.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *CONLL 2002*.

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

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Osamu Imaichi, and Tomoaki Imamura. 1997. Japanese morphological analysis system ChaSen manual. Technical Report NAIST-IS-TR97007, Nara Institute of Science and Technology. In Japanese.

Yoshihiko Nitta, Atushi Okajima, Hiroyuki Kaji, Youichi Hidano, and Koichiro Ishihara. 1984. A proper treatment of syntax and semantics in machine translation. In *COLING 84*, pages 159–166.

Atsushi Okazaki, Kiyoaki Shirai, Takenobu Tokunaga, and Hozumi Tanaka. 2001. A syntactic annotation tool with user navigation. In *15th Annual Conference of Japanese Society for Artificial Intelligence*. In Japanese.

Kiyoaki Shirai, Takenobu Tokunaga, and Hozumi Tanaka. 1995. Automatic extraction of Japanese grammar from a bracketed corpus. In *Natural Language Processing Pacific Rim Symposium*, pages 211–216.

Kiyoaki Shirai, Masahiro Ueki, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. 2000. MSLR parser – tools for natural language analysis. *Journal of Natural Language Processing*, 7(5):93–112. In Japanese.

Kiyotaka Uchimoto, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara. 2000. Dependency model using posterior context. In *6th International Workshop on Parsing Technologies*.