A Large-Scale Japanese CFG Derived from a Syntactically Annotated Corpus and Its Evaluation

Tomoya Noro Taiichi Hashimoto
Takenobu Tokunaga Hozumi Tanaka

Tokyo Institute of Technology
Graduate School of Information Science and Engineering
{noro,taiichi,take,tanaka}@cl.cs.titech.ac.jp

1 Introduction

Although large-scale grammars are prerequisite for parsing a great variety of sentences, 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 (tree-bank grammar) can parse sentences with high accuracy and coverage although the method for deriving a CFG is very simple [1]. Actually, there have been quite a few studies concerning this kind of grammars. For Japanese, however, CFGs cannot be derived using the Charniak's method since there is no large-scale syntactically annotated corpus such as the Penn Treebank corpus [1]. Therefore such corpus needs to be developed to enable derivation of a large-scale CFG.

However, even if a large-scale, syntactically annotated corpus were already available, a CFG derived from it can be unsatisfactory, in as it creates a great number of possible parses (in average more than 10^{12}, according to our preliminary experiment). Too many parse results do not only reduce the parsing accuracy and parsing speed, but also require larger memory to parse and store long sentences. Although Charniak has removed some CFG rules (e.g. rules occurring only once

[1] The EDR Japanese corpus [4] is one of the large-scale Japanese corpora. However, unlike the Penn Treebank corpus, it is a bracketed corpus (nonterminal symbols are not assigned to each intermediate node). Shirai et al. proposed a method to derive a CFG from the EDR corpus, guessing nonterminal symbols to be assigned automatically to each intermediate node using some heuristics [17].

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” structure, the derived grammar creates many parse results, representing a different possible reading, i.e. meaning. A syntactic parser does not deal with semantics. Hence, it is difficult to deal with ambiguity of that sort. On the other hand, if the parser creates many different parses, it becomes difficult to disambiguate the results, even if semantic analysis is carried out after the syntactic parsing. We assume that syntactic analysis based on a large-scale CFG is followed by semantic analysis. Since the parse results are sent to the subsequent semantic processing, the number of parse results should be as small as possible. Therefore, it is necessary to build a CFG that minimizes the ambiguity during the 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 causes of 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 and time-consuming, 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 for building such a large-scale Japanese CFG, under the assumption that the parse results will be subsequently sent to the semantic processing module. We also provide an experimental evaluation of the obtained CFG showing reduction in the number of parse results (reduced ambiguity) created by the CFG and the improved parsing accuracy. Several methods for tree transformation have been proposed for other languages [10, 16]. Although our work is similar, the difference is that we consider parsing ambiguity as well as parsing accuracy. 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 Ambiguity

To decrease the ambiguity (i.e the number of parse results), we start by analyzing main causes. There are four main causes of ambiguity:

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

Inconsistency: There may be contradiction concerning the structure since large-scale corpora are usually built incrementally and by several annotators. A CFG derived from an inconsistent corpus can yield many parse results with inconsistent structures.

Lack of Syntactic Information: Some important syntactic information might be lost during the CFG derivation since CFG rules generally represent only structures of subtrees of depth one (relation between a parent node and some child node). Yet, in case of Japanese, a verb phrase can be an adnominal phrase, continuous clause, or subordinate clause. In order to decide which one to choose, one has to consider verb conjugation or 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* (wearing a hat)" could be an adnominal phrase, because the conjugation of the verb "*kabutteiru* (wear)" is an adnominal form. If no information concerning verb conjugation can be assigned at intermediate nodes of the subtree covering the verb phrase, it is not clear whether the verb phrase is an adnominal phrase or continuous clause.

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

Since the first and second causes are types of annotation errors, they need to be corrected manually as soon as they are found[2]. On the other hand, since the third and fourth causes 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.

3 Policy for Modifying the Corpus and the CFG

In order to avoid the third cause of ambiguity, information should be added to each intermediate node in the structure, where necessary. On the other hand, some am-

[2] Although this kind of error can be automatically corrected (or detected) in some methods [2, 3], not all of them can be corrected. They should be manually corrected at the end. Furthermore, Japanese has another problem that English does not have: there are potential errors in word segmentation since words are not separated by spaces.

biguity due to the fourth cause should be left to the subsequent semantic processing since it is difficult to reduce the ambiguity without recourse to semantic information during syntactic parsing. This can be achieved by representing the ambiguous cases as the same structure.

We have considered modification for verb conjugation, compound noun structure, adverbial and adnominal phrase attachment and conjunctive structure. In this section, we describe their modification briefly. The details are given in [14].

3.1 Verb Conjugation

As mentioned in the previous section, information of verb conjugation or particles (postpositions) at the end of the verb phrase is important to judge whether the phrase should be adverbial phrase or adnominal phrase. We add the information to each intermediate node related to the verb (cf. "SPLIT-VP" in [10] and "Verb Form" in [16]).

3.2 Compound Noun Structure

In general, it is difficult to disambiguate structure of compound noun without any semantic information. Shirai et al. modify their CFG to produce a right linear binary branching tree for compound nouns during the parse [17][3]. We modify the structure in the same way: structure ambiguity of compound noun is represented as the same structure regardless of the meaning or word-formation.

3.3 Adverbial and Adnominal Phrase Attachment

Semantic information is necessary to disambiguate adverbial and adnominal phrase attachment. However, it is meaningless to represent all of the ambiguity as the same structure regardless of the meaning, since it means no decision about phrase attachment is made during syntactic parsing and it makes the subsequent semantic processing difficult. Some of the ambiguity should be represented as the different structure (i.e. the ambiguity is unresolved during syntactic parsing). We represent structure ambiguity of adnominal phrase attachment as the same structure regardless of the meaning while we distinguish structure ambiguity of adverbial phrase attachment by meaning. In case of a phrase like "*watashi no chichi no hon* (my father's book)", the structure is same whether the adnominal phrase "*watashi no*

[3] 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 (POS) 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.)

(my)” attaches to the noun “*chichi* (father)” or the noun “*hon* (book)”. On the other hand, in case of a sentence like “*kare ga umi wo egaita e wo katta*”, we distinguish the structure according to whether the adverbial phrase “*kare ga* (he)” attaches to the verb “*egaita* (paint)” (it means “I bought a picture of a sea painted by him”) or the verb “*katta* (buy)” (it means “he bought a picture of a sea”).

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 adnominal phrase [4]. We have also decided to annotate a corpus as described above since adverbial phrase attachment can be disambiguated in some cases using syntactic information (e.g. particles, punctuation).

3.4 Conjunctive Structure

In general, parsing accuracy of the sentences containing conjunctive structures is significantly worse than that of sentences without such structures. Our preliminary experiments show that the sentence accuracy of such sentences is only about half of the rest [5]. 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 [9]. Contrary to their method, our CFG does not specify conjunctive structures during syntactic parsing, as they are assumed to be analyzed during the subsequent semantic processing (similar to “Coordinated Categories” in [16]).

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 created by the derived CFG, and the accuracy of the parsing achieved when using the CFG.

[4] There are cases where this discrimination is not so easy. For instance, 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.

[5] The definition of sentence accuracy is described later.

As mentioned earlier, it is important to decrease the number of parse results, as this speeds up the processing 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. We evaluated on the EDR Japanese corpus [4] (we refer to this corpus as "EDR corpus") and the RWC corpus [5].

4.1 Evaluation on the EDR corpus

The EDR corpus is a bracketed corpus with only skeletal structures recorded for each sentences. The intermediate nodes of the structure are not assigned with non-terminal symbols. We extracted 8,911 sentences (on average 20.01 morphemes in a sentence) from it and manually annotated "semantically correct" structure of each sentence (we refer to this corpus as "EDR original corpus"). Then we modified the structure according to the policy described above by an annotation tool [15] (we refer to this corpus as "EDR modified corpus").

We followed bracket structure in the EDR corpus to annotate "EDR original corpus". Since POS system of the EDR corpus is so coarse (only 15 POS tags), we assigned fine-grained POS tags from the EDR Japanese word dictionary. Each word in this dictionary has left and right adjacency attribute (i.e. information about what kind of POS tag can precede or follow the word) and surface case information for verbs and adjectives (i.e. information about what kind of case the verb or adjective takes) [6] as well as POS tag. We combined them and used as the POS tag set for the both EDR original corpus and EDR modified corpus.

CFGs are derived from the original and modified corpus (we refer to these two CFGs as "EDR original CFG" and "EDR modified CFG" respectively), and used to parse POS sequences of sentences in the corpus by MSLR parser [18] [7]. The number of rules in two CFGs and the number of parse results are shown in Table 1. The number of parse results decreased by 10^7 order, while the number of CFG rules increased by 255 [8].

Next, we ranked parse results by training the parser according to the probabilistic generalized LR (PGLR) model [6] using 10-fold cross-validation (CFGs were derived from the training data only). We examined three kinds of evaluation metrics:

$$\text{Coverage} = 1 - \frac{\text{\# sentences failed in parsing}}{\text{\# all sentences}}$$

[6] In case of English, it is similar to information about whether the verb is intransitive, transitive or ditransitive.

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

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

Table 1: The number of CFG rules and the number of parse results (EDR original CFG vs EDR modified CFG)

	# CFG rules	# non-terminals	# terminals	# parse results
EDR (original)	1,694	249	600	1.868×10^{12}
EDR (modified)	1,949	279	600	9.355×10^{5}

Table 2: Coverage and recall (EDR original CFG vs EDR modified CFG)

	Coverage	Recall
EDR (original)	98.51%	96.63%
EDR (modified)	97.32%	95.88%

$$\text{Recall} = \frac{\text{\# sentences parsed correctly}}{\text{\# all sentences}}$$

$$\text{Sentence Accuracy} = \frac{\text{\# sentences parsed correctly in the top-}n\text{ parse results}}{\text{\# all sentences}}$$

"Sentences failed in parsing" means no result can be created in parsing the sentences. "Sentences parsed correctly" means the sentences in which all constituents are labeled correctly (i.e. exact match) in all the parse results, and "Sentences parsed correctly in the top-n parse results" means the sentences in which all constituents are labeled correctly in the top-n parse results ranked by PGLR model. Since the parse results are re-analyzed using semantic information in the subsequent processing, the structure of the parse result must match the correct structure exactly. That is why we use this evaluation metric rather than labeled precision and labeled recall, which are commonly used in evaluation of parsing.

Results are shown in Table 2 and Figure 1. Coverage and recall decreased by around 1%. Despite the decrease of coverage and recall, sentence accuracy increased about 8% under assumption that the top-100 parse results are re-analyzed in the subsequent processing. On the other hand, only the top-10 parse results are enough for the EDR modified CFG to overcome the accuracy among top-100 parse results using the EDR original CFG.

Some readers might take it for granted that sentence accuracy increases if the EDR modified corpus is used as a gold-standard because certain difficult decisions are not made in annotation and left to the subsequent processing. To test the accuracy if the EDR original corpus is used as a gold-standard, we randomly selected 100 sentences from the EDR modified corpus and examined dependency accuracy (the percentage of correct dependency relations out of all dependency relations) of the top parse results ranked by PGLR model (the EDR original corpus is used as a gold-standard). Since phrase structure is annotated in the corpus and the

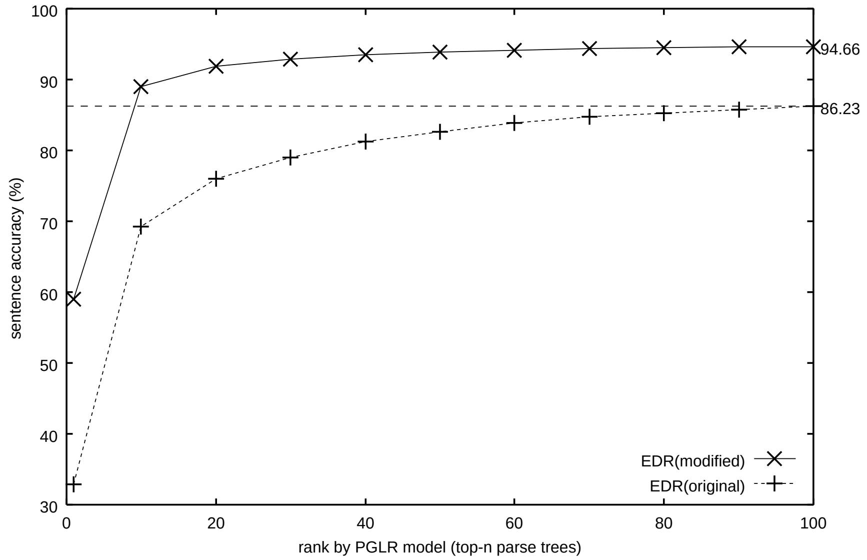

Figure 1: Sentence accuracy (EDR original CFG vs EDR modified CFG)

EDR modified CFG does not create dependency structures but phrase structures, we converted the parse results and structures in the EDR original corpus to dependency structures. Since the CFG does not determine adnominal phrase attachment, we assume that every ambiguous adnominal phrase attaches to the nearest noun. Whether the relation between two units is conjunctive or not is distinguished in this evaluation. 96 sentences were correctly segmented into Japanese phrasal units (*bunsetsu*), and dependency accuracy was 89.23%, which rivals the state-of-the-art dependency analysis using support vector machine, maximum entropy, etc [7, 8, 19] [9] although no semantic information is considered yet. We expect that the accuracy will increase as soon as semantic information is incorporated in the subsequent processing. The method of incorporating semantic information is left for future research.

4.2 Evaluation on the RWC corpus

There is a problem with using the CFG derived from the EDR corpus: there is no morphological analyzer based on the POS system used in this corpus. Thus we evaluated on the RWC corpus, a tagged corpus whose POS system is based on the Japanese morphological analyzer, ChaSen [12]. We extracted 16,421 sentences (on

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

average 21.71 morphemes in each sentence) from it and we annotated the “modified corpus” only without annotating the “original corpus” according to [11]. We refer the CFG derived from the corpus as “RWC CFG”.

The POS system of the RWC corpus (i.e. the POS system of ChaSen) is not sufficient for syntactic parsing. For instance, particles (postpositions) should be classified by word [17]. Some word sequences such as phrases which act as auxiliary verbs should be merged to reduce unnecessary ambiguity. We convert POS tags in the RWC corpus automatically before annotating. The main changes in POS tags are follows:

1. Numeral sequences are merged (ChaSen splits numeral sequences along characters).

2. Case particles are classified by case.

3. Verb endings for past tense (e.g. “*ta*”), gerund (e.g. “*te*”) and others (e.g. “*tara*”, “*tari*”) [10] are merged with the previous verb [11].

4. Sequences of alphabet (i.e. roman) characters are labeled as common noun.

5. Word sequences which act as auxiliary verbs (e.g. “*noda*”) are merged.

6. Suffixes for changing nouns to verbs (e.g. “*suru*”) are separated from other verbs.

7. Symbols which are usually used at the end of sentences (e.g. question mark) are separated from other symbols [17].

8. Adverbs which are also used as noun modifiers are separated from other adverbs (similar to “Adverbial Classification” in [16]).

9. The latter verbs of verb sequences in [13] are labeled as auxiliary verb. For instance, a verb sequence “*fuki kesu* (blow out)” consists of two verbs “*fuku* (blow)” and “*kesu* (put out)”, and the latter verb “*kesu*” is labeled as auxiliary verb.

We evaluated on the corpus and the CFG derived from it in the same way as we did for the EDR corpus. Results are shown in Table 3, Table 4 and Figure 2. The number of parse results was 9.599×10^4, coverage and recall were 98.38% and 97.18% respectively, and sentence accuracy among top-100 parse results was 95.76%. These results are comparable to the evaluation on the EDR corpus[11].

[10] This type of verb ending is called “*ta*-series ending” [11].

[11] We have not examined dependency accuracy, since we did not annotate the “original corpus”.

Table 3: The number of CFG rules and the number of parse results (EDR modified CFG vs RWC CFG)

	# CFG rules	# non-terminals	# terminals	# parse results
EDR (modified)	1,949	279	600	9.355×10^5
RWC	2,556	290	391	9.599×10^4

Table 4: Coverage and recall (EDR modified CFG vs RWC CFG)

	Coverage	Recall
EDR (modified)	97.32%	95.88%
RWC	98.38%	97.18%

5 Conclusion

Although a large-scale CFG can be derived from a syntactically annotated corpus, in general, such CFGs create a large number of parse results. The principal cause is due to the fact that such CFGs are not built so as to sufficiently limit the ambiguity. We show that a practical large-scale CFG for syntactic parsing can be built by investigating the cause of increased ambiguity and modifying a corpus and consequently a CFG to remove the cause of such ambiguity.

Since we assume that the parse results created by our CFG are re-analyzed in the subsequent processing, we have to provide a method for re-analysis of the parse results. Our policy for annotating a corpus has been considered with several types of ambiguity: structure of compound noun, adnominal phrase attachment, adverbial phrase attachment and conjunctive structure. We are planning to provide each method individually and integrate them into one processing.

References

[1] Charniak, Eugene (1996) Tree-bank Grammars. In *the 13th National Conference on Artificial Intelligence*, pp. 1031–1036.

[2] Dickinson, Markus and Meurers, W. Detmar (2003) Detecting Errors in Part-of-Speech Annotation. In *the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*.

[3] Dickinson, Markus and Meurers, W. Detmar (2003) Detecting Inconsistencies in Treebanks. In *the 2nd Workshop on Treebank and Linguistic Theories (TLT 2003)*.

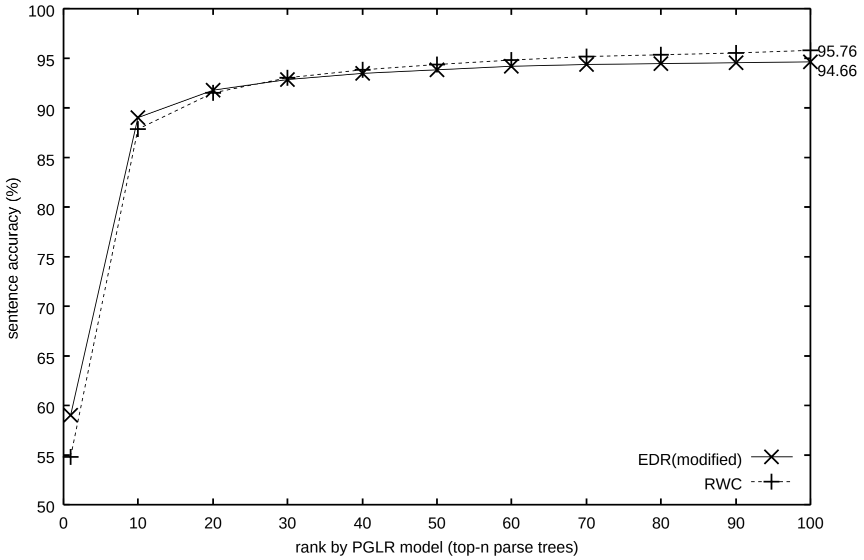

Figure 2: Sentence accuracy (EDR modified CFG vs RWC CFG)

[4] EDR (1994) EDR Electronic Dictionary User's Manual, 2.1 edition. In Japanese.

[5] Hashida, Koichi, Isahara, Hitoshi, Tokunaga, Takenobu, Hashimoto, Minako, Ogino, Shiho and Kashino, Wakako (1998) The RWC Text Databases. In *the 1st International Conference on Language Resource and Evaluation (LREC 1998)*, pp. 457–461.

[6] Inui, Kentaro, Sornlertamvanich, Virach, Tanaka, Hozumi and Tokunaga, Takenobu (2000) Probabilistic GLR parsing. In Bunt, Harry and Nijholt, Anton (eds) *Advances in Probabilistic and Other Parsing Technologies*, pp. 85–104. Kluwer Academic Publishers.

[7] Kanayama, Hiroshi, Torisawa, Kentaro, Mitsuishi, Yutaka and Tsujii, Jun'ichi (2000) A Hybrid Japanese Parser with Hand Crafted Grammar and Statistics. In *the 18th International Conference on Computational Linguistics (COLING 2000)*, pp. 411–417.

[8] Kudo, Taku and Matsumoto, Yuji (2002) Japanese Dependency Analysis Using Cascaded Chunking. In *Conference on Computational Natural Language Learning (CoNLL 2002)*.

[9] Kurohashi, Sadao and Nagao, Makoto (1994) A Syntactic Analysis Method of Long Japanese Sentences Based on the Detection of Conjunctive Structures. *Computational Linguistics*, 20(4), pp. 507–534.

[10] Klein, Dan and Manning, Christopher D. (2003) Accurate Unlexicalized Parsing. In *the 41st Annual Meeting of Association for Computational Linguistics (ACL 2003)*, pp. 423–430.

[11] Masuoka, Takashi and Takubo, Yukinori (1992) *Kiso Nihongo Bunpou (Foundation of Japanese Grammar)*, Kurosio Shuppan. In Japanese.

[12] Matsumoto, Yuji, Kitauchi, Akira, Yamashita, Tatsuo, Hirano, Yoshitaka, Matsuda, Hiroshi, Takaoka, Kazuma and Asahara, Masayuki (2000) *Japanese Morphological Analysis System ChaSen version 2.2.1 Manual*. Nara Institute of Science and Technology.

[13] Nomura, Masaaki and Ishii, Masahiko (1987) *Fukugoudoushi Shiryoushuu (Collection of Data about Compound Verbs)*, National Institute for Japanese Language. In Japanese.

[14] Noro, Tomoya, Hashimoto, Taiichi, Tokunaga, Takenobu and Tanaka, Hozumi (2004) Building a Large-Scale Japanese CFG for Syntactic Parsing. In *the 4th Workshop on Asian Language Resources (ALR 2004)*, pp. 71–78.

[15] Okazaki, Atsushi, Shirai, Kiyoaki, Tokunaga, Takenobu and Tanaka, Hozumi (2001) A Syntactic Annotation Tool with User Navigation. In *the 15th Annual Conference of Japanese Society for Artificial Intelligence*. In Japanese.

[16] Schiehlen, Michael (2004) Annotation Strategies for Probabilistic Parsing in German. In *the 20th International Conference on Computational Linguistics (COLING 2004)*, pp. 390–396.

[17] Shirai, Kiyoaki, Tokunaga, Takenobu and Tanaka, Hozumi (1995) Automatic Extraction of Japanese Grammar from a Bracketed Corpus. In *Natural Language Processing Pacific Rim Symposium*, pp. 211–216.

[18] Shirai, Kiyoaki, Ueki, Masahiro, Hashimoto, Taiichi, Tokunaga, Takenobu and Tanaka, Hozumi (2000) MSLR Parser – Tools for Natural Language Analysis. *Journal of Natural Language Processing*, 7(5), pp. 93–112. In Japanese.

[19] Uchimoto, Kiyotaka, Murata, Masaki, Sekine, Satoshi and Isahara, Hitoshi (2000) Dependency Model Using Posterior Context. In *the 6th International Workshop on Parsing Technologies (IWPT 2000)*.