BUILDING A LARGE-SCALE JAPANESE SYNTACTICALLY ANNOTATED CORPUS FOR DERIVING A CFG

Tomoya Noro, Taiichi Hashimoto, Takenobu Tokunaga and Hozumi Tanaka

Graduate School of Information Science and Engineering, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552, Japan
E-mail: {noro,taiichi,take,tanaka}@cl.cs.titech.ac.jp

ABSTRACT

A syntactically annotated corpus is a type of language resources, which can be used in many different ways. One possible use of the corpus is deriving a context-free grammar (CFG), and there have been quite a few studies concerning this type of grammar derivation. However, a CFG derived from a syntactically annotated corpus often has a shortcoming: it creates a great number of parse results (ambiguity) during syntactic parsing. We have been building a Japanese syntactically annotated corpus in order to derive a grammar with less ambiguity during syntactic parsing. In this paper, we introduce our policy for building the corpus and an experimental evaluation of the derived CFG.

1. INTRODUCTION

A syntactically annotated corpus is a type of language resources, which is used for obtaining statistical information concerning corpus-based NLP technologies. Another use of the corpus is deriving a CFG, and there have been quite a few studies concerning this kind of grammar. For English, it is well known that a CFG derived from the Penn Treebank corpus (tree-bank grammar) [1] can parse sentences with high accuracy and coverage although the method for deriving CFG is very simple [2]. Several corpora like the Penn Treebank corpus have been built for other languages and many algorithms (grammar derivation, syntactic parsing, etc.) have been proposed. However, such corpus has not been built for Japanese yet. Therefore such Japanese corpus needs to be built to apply the algorithms to Japanese.

However, even if a syntactically annotated corpus were already available, a CFG derived from it can be unsatisfactory, in as it creates a great number of possible parses (large ambiguity). 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 in the Penn Treebank corpus) to avoid such problems [2], this is not enough, as the rules that occur more than once may also increase ambiguity.

Since the sentences of an ordinary syntactically annotated corpus have "semantically correct" structure, the derived CFG creates many parse results, whereas each result represents different possible readings, i.e. meanings. A syntactic parser does not deal with semantics. Hence, it is difficult to deal with semantic ambiguity. 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 corpus so that the derived CFG would create less parse results during syntactic parsing.

We attempt to build such a corpus by using the following method: (1) derive a CFG from an existing corpus, (2) analyze major causes of ambiguity, (3) create a policy for modifying the corpus, (4) modify the corpus according to the policy and re-derive a CFG from it, and (5) repeat steps (2) - (4) until most problems are solved. While the step (5) is labor-intensive and time-consuming, it is very important to do so in order to build a large-scale corpus and to derive CFG for syntactic parsing from it.

We have been building a Japanese corpus so that the derived CFG would create less ambiguity during syntactic parsing [3, 4]. In this paper, we introduce our policy for building the corpus, and show an experimental evaluation of the derived CFG. Several methods for tree transformation have been proposed for other languages, such as English [5] and German [6]. Although our work is similar, the difference is that we consider parsing ambiguity as well as parsing accuracy.

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 [3]:

1. **Human Errors:** Human annotators sometimes make mistakes when annotating syntactic structure.

2. **Inconsistency:** There may be contradiction concerning the structure since large-scale corpora are usually built incrementally and by several annotators.

3. **Lack of Syntactic Information:** Some syntactic information which is important for syntactic parsing might be lost during the CFG derivation.

4. **Need for Semantic Information:** Semantic information is necessary for disambiguation in some cases.

Since the first and second causes are types of annotation errors, they need to be corrected manually as soon as they are found [1]. 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 CFG from this newly-annotated corpus.

3. POLICY FOR MODIFYING THE CORPUS

In order to avoid the third cause of ambiguity, syntactic information should be added to each intermediate node in the structure, where necessary. On the other hand, ambiguity due to the fourth is better be left to the subsequent semantic processing since it is difficult to reduce such 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, conjunctive structure [3, 4].

Verb Conjugation: We add information of verb conjugation to each intermediate node related to the verb (cf. "SPLIT-VP" in [5] and "Verb Form" in [6]).

Compound Noun Structure: Structure ambiguity of compound noun is represented as the same structure (right linear binary branching tree) regardless of the meaning or word-formation (cf. "X-*Retu*" in [9]).

Adnominal Phrase Attachment: Structure ambiguity of adnominal phrase attachment (e.g. whether the adnominal phrase "*watashi no* (my)" attaches to the noun "*chichi* (father)" or "*hon* (book)" in case of a phrase "*watashi no chichi no hon* (my father's book)") is represented as the same structure (right linear binary branching tree).

Adverbial Phrase Attachment: Structure ambiguity of adverbial phrase attachment (e.g. whether the adverbial phrase "*tobira wo* (door)" attaches to the verb "*akete* (opened)" or "*haitta* (entered)" in case of a phrase "*tobira wo akete heya ni haitta* (I opened the door and entered the room)") is distinguished by meaning.

Conjunctive Structure: Conjunctive structure is not specified in syntactic structure (cf. "Coordinated Categories" in [6]).

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 is left as is during syntactic parsing. However, this increase of ambiguity is not significant. Actually, in Japanese it is relatively easy to discriminate between an adverbial and adnominal phrase. 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).

4. EVALUATION

To evaluate the efficiency of the corpus modified according to our policy and the CFG derived from the corpus, we consider two aspects: the number of parse results created by the derived CFG, and the accuracy of the parsing achieved when using the CFG. We evaluated on the EDR corpus [10] and the RWC corpus [11].

4.1. Evaluation on the EDR corpus

The EDR corpus is a bracketed corpus with only skeletal structures recorded for each sentence (non-terminal symbols are not assigned to each intermediate node of the structure). We extracted 8,911 sentences from the corpus and manually annotated "semantically correct" structure of each sentence (we refer to this corpus as "EDR original corpus") [2]. Then we modified the structure according to the policy described above by an annotation tool [12] to obtain "EDR modified corpus".

CFGs are derived from the EDR original corpus and the EDR modified corpus ("EDR original CFG" and "EDR modified CFG" respectively), and used to parse POS sequences of sentences in the corpus by MSLR parser [13]. The number of CFG 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 only 255.

Next, we ranked parse results by training the parser according to probabilistic generalized LR (PGLR) model [14]

[1] Several methods for correcting (or detecting) this kind of error have been proposed [7, 8].

[2] We followed the bracket structure in the EDR corpus to annotate "EDR original corpus".

Table 1. The number of parse results

	# CFG rules	# parse results
EDR (original)	1,694	1.868×10^{12}
EDR (modified)	1,949	9.355×10^{5}
RWC (modified)	2,565	9.599×10^{4}

Table 2. Coverage and recall

	Coverage	Recall
EDR (original)	98.51%	96.63%
EDR (modified)	97.32%	95.88%
RWC (modified)	98.38%	97.18%

using 10-fold cross-validation (CFGs were derived from the training data only). We examined three evaluation metrics:

Coverage: The percentage of sentences where at least one parse result can be obtained in parsing

Recall: The percentage of sentences where the correct parse (i.e. exact match) can be obtained in parsing

Sentence Accuracy: The percentage of sentences where the correct parse is in the top-n parse results ranked by PGLR model.

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 by about 8% under assumption that the top-100 parse results are reanalyzed in the subsequent processing (i.e. when $n = 100$).

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 are 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 three evaluation metrics:

Segmentation Accuracy (SegA): The percentage of sentences which are correctly segmented into *bunsetsu* (Japanese phrasal unit).

Dependency Accuracy (DepA): The percentage of correct dependency relations out of all dependency relations

Sentence Accuracy (SenA): The percentage of sentences where all dependency relations are correct

Since phrase structure is annotated in the corpus and the EDR modified CFG does not create dependency structure but phrase structure, we converted the parse results and structures in the EDR original corpus to dependency structures

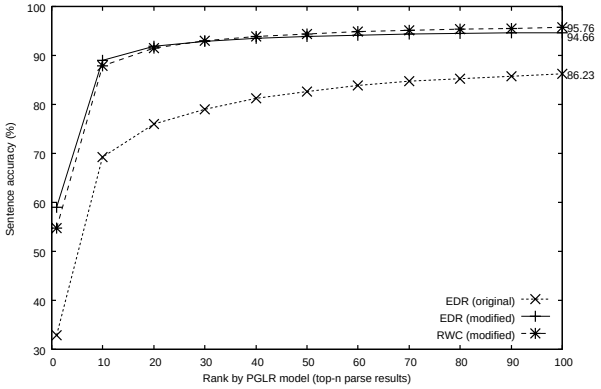

Fig. 1. Sentence accuracy

Table 3. Dependency accuracy

	SegA	DepA	SenA
EDR	91.00%	91.32%	61.54%
RWC	65.99%	85.76%	52.38%

[3]. Results are shown in table 3. Dependency Accuracy was 91.32%, which rivals the state-of-the-art dependency analysis using KNP [15] (89.97%) [4], support vector machine [17] (89.29%), maximum entropy [18] (87.93%), etc [5] although no semantic information is incorporated in the subsequent processing. We expect that the accuracy will increase as soon as semantic information is incorporated in the subsequent processing.

4.2. Evaluation on the RWC corpus

To examine whether our policy can be applicable to other corpora, we evaluated on the RWC corpus [11], a POS tagged corpus (no syntactic structure) [6], in the same way as we did for the EDR corpus. We extracted 16,421 sentences from the RWC corpus and annotated the "RWC modified corpus" only without annotating the "RWC original corpus". We refer the CFG derived from the modified corpus as "RWC modified CFG". Results are shown in table 1, table 2, figure 1, and table 3. Since we did not prepare the RWC original corpus, we used the Kyoto corpus [19] (3,764 sentences) as a gold-standard when evaluating dependency accuracy [7]. While the number of parse results (table 1), coverage, recall

[3]We assume that every ambiguous adnominal phrase attaches to the nearest noun. Whether the relation between two units is conjunctive or not is not distinguished in this evaluation.

[4]Dependency accuracy using KNP is quoted from [16].

[5]The comparison are not absolutely fair since the results are for different corpora.

[6]The POS system of the RWC corpus is different from that of the EDR corpus

[7]Since the POS system of the Kyoto corpus is different from that of the RWC corpus, we converted POSs automatically.

(table 2) and sentence accuracy (figure 1) are comparable to the evaluation on the EDR corpus, dependency accuracy decreased. However, we expect that the accuracy will increase as soon as semantic information is incorporated in the subsequent processing [8].

5. CONCLUSION AND FUTURE WORK

In this paper, we introduced our policy for syntactically annotating a Japanese corpus so that the derived CFG would create less parse results during syntactic parsing. Our policy was applied to and evaluated on the EDR and the RWC corpora. Results show that the number of parse results the derived CFG creates can be decreased by adhering to our policy, and that our policy can easily be applied to other corpora.

Since we assume that the parse results created by our CFG are re-analyzed in the subsequent processing, in the future, we have to provide a method for semantic analysis of the obtained parse results.

6. REFERENCES

[1] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[2] Eugene Charniak, "Tree-bank grammars," in *AAAI-96*, 1996, pp. 1031–1036.

[3] Tomoya Noro, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka, "Building a large-scale Japanese CFG for syntactic parsing," in *ALR 2004*, 2004, pp. 71–78.

[4] Tomoya Noro, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka, "A large-scale Japanese CFG derived from a syntactically annotated corpus and its evaluation," in *TLT 2004*, 2004, pp. 115–126.

[5] Dan Klein and Christopher D. Manning, "Accurate unlexicalized parsing," in *ACL 2003*, 2003, pp. 423–430.

[6] Michael Schiehlen, "Annotation strategies for probabilistic parsing in German," in *COLING 2004*, 2004, pp. 390–396.

[7] Markus Dickinson and W. Detmar Meurers, "Detecting errors in part-of-speech annotation," in *EACL 2003*, 2003.

[8] According to our preliminary experiment, dependency accuracy would increase up to 94% if the subsequent processing could choose the best parse result among the top-100 parse results and analyze adnominal phrase attachment in the best way.

[8] Markus Dickinson and W. Detmar Meurers, "Detecting inconsistencies in treebanks," in *TLT 2003*, 2003.

[9] Kiyoaki Shirai, Takenobu Tokunaga, and Hozumi Tanaka, "Automatic extraction of Japanese grammar from a bracketed corpus," in *NLPRS 95*, 1995, pp. 211–216.

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

[11] Koichi Hashida, Hitoshi Isahara, Takenobu Tokunaga, Minako Hashimoto, Shiho Ogino, and Wakako Kashino, "The RWC text databases," in *LREC 98*, 1998, pp. 457–461.

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

[13] Kiyoaki Shirai, Masahiro Ueki, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka, "MSLR parser – tools for natural language analysis," *Journal of Natural Language Processing*, vol. 7, no. 5, pp. 93–112, 2000, In Japanese.

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

[15] Sadao Kurohashi and Makoto Nagao, "KN Parser: Japanese dependency / case structure analyzer," in *the International Workshop on Sharable Natural Language Resources*, 1994, pp. 48–55.

[16] Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara, "Japanese dependency structure analysis based on maximum entropy models," *Information Processing Society of Japan*, vol. 40, no. 9, pp. 3397–3407, 1999, In Japanese.

[17] Taku Kudo and Yuji Matsumoto, "Japanese dependency analysis using cascaded chunking," in *CONLL 2002*, 2002.

[18] Kiyotaka Uchimoto, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara, "Dependency model using posterior context," in *IWPT 2000*, 2000.

[19] Sadao Kurohashi and Makoto Nagao, "Kyoto university text corpus project," in *ACL 97*, 1997, pp. 115–118, In Japanese.