

Semi-Automatic Construction of a Tree-Annotated Corpus Using an Iterative Learning Statistical Language Model

Kiyoaki Shirai, Hozumi Tanaka, Takenobu Tokunaga

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

Abstract

In this paper, we propose a method to construct a tree-annotated corpus, when a certain statistical parsing system exists and no tree-annotated corpus is available as training data. The basic idea of our method is to sequentially annotate plain text inputs with syntactic trees using a parser with a statistical language model, and iteratively retrain the statistical language model over the obtained annotated trees. The major characteristics of our method are as follows: (1) in the first step of the iterative learning process, we manually construct a tree-annotated corpus to initialize the statistical language model over, and (2) at each step of the parse tree annotation process, we use both syntactic statistics obtained from the iterative learning process and lexical statistics pre-derived from existing language resources, to choose the most probable parse tree.

1. Introduction

In recent years, many researchers have been devoted time to the study of statistical parsing (Charniak, 1997; Collins, 1996; Li, 1996). In general, statistical parsing is a technique by which a parsing system ranks parse trees and chooses the most probable one according to syntactic statistics such as structural preferences, trained from a corpus. For supervised learning of syntactic statistics, tree-annotated corpora such as the Penn Treebank (Marcus et al., 1993) are needed. Several Japanese tree-annotated corpora have also been developed, including the EDR corpus (EDR, 1995) and Kyoto University corpus (Kurohashi and Nagao, 1997). Thus a considerable amount of literature exists relating to the statistical parsing of Japanese sentences (Fujio and Matsumoto, 1998; Inui and Inui, 2000; Uchimoto et al., 1999).

However, existing tree-annotated corpora often cannot be applied to training syntactic statistics, because grammar used in statistical parsing system differs from the one underlying the tree-annotated corpora. Many researchers have statistical parsers with individuated grammars, but a corpus in which parse tree annotation is based on the same grammar, is not always available. In addition to this, differences in part-of-speech (POS hereafter) tag sets also pose a problem. When the POS tag set of the tree-annotated corpus differs from the one used in the lexicon of the parsing system, syntactic statistics from the corpus cannot be applied directly in the parser. For the syntactic analysis of unsegmented languages such as Japanese, agreement of word segmentation schemata is also a prerequisite of being able to use existing corpora as training data. Therefore, when a certain statistical parsing system exists and no tree-annotated corpus is available as training data, it is necessary to newly construct a tree-annotated corpus based on the grammar and lexicon of the parser.

In this paper, we propose a method to construct a tree-annotated corpus. The basic idea of our method is to sequentially annotate plain text inputs with syntactic trees using a parser with a statistical language model, and iteratively retrain the statistical language model over the obtained annotated trees. The iterative learning process of the statistical language model is similar to the Inside-Outside algorithm (Lari and Young, 1990), a learning algorithm used to train the parameters of a PCFG. The major characteristics of our method are as follows:

- In the first step of the iterative learning process, we manually construct a tree-annotated corpus to initialize the statistical language model over. The size of this corpus is small so as to reduce human effort.
- At each step of the parse tree annotation process, we use both syntactic statistics obtained from the iterative learning process and lexical statistics pre-derived from existing language resources, to choose the most probable parse tree.

In what follows, we first briefly introduce our statistical language model in Section 2. We next describe details of the iterative parse tree annotation process in Section 3. We also present our method for obtaining lexical statistics. We then describe the result of an experiment to evaluate the effectiveness of our method in Section 4. Finally, we conclude this paper and discuss future work in Section 5.

2. Statistical Language Model

In this section, we describe our statistical language model (Shirai et al., 1998) in brief.

As with most statistical parsing frameworks, given an input string A , we rank its parse trees according to the joint distribution $P(R, W)$, where W is a word sequence candidate for A , and R is a parse tree candidate

for W whose terminal symbols constitute a POS tag sequence L . Figure 1 shows an example of a parse tree for the Japanese sentence “kanojo ga mado o ake ta” (She opened the window). We first decompose $P(R, W)$ into

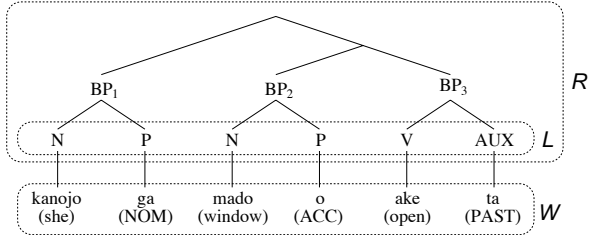


Figure 1: Example of a Parse Tree for the Sentence “kanojo ga mado o ake ta” (She opened the window).

three submodels, the syntactic model $P(R)$, the derivation model $P(W|L)$ and the lexical dependency model $D(W|R)$:

$$P(R, W) = P(R) \cdot P(W|L) \cdot D(W|R) \quad (1)$$

The first submodel is the syntactic model $P(R)$, which is the generation probability of parse tree R and reflects syntactic statistics such as structural preferences. At present, we estimate $P(R)$ using the probabilistic GLR (PGLR) language model, which is founded on the incorporation of probabilistic distributions into the GLR parsing framework (Inui et al., 1997; Sornlertlamvanich et al., 1997). The PGLR language model can reflect mild context-sensitiveness and is easily trainable given a tree-annotated corpus.

The second submodel is the derivation model $P(W|L)$, which reflects word occurrence statistics. It is given by the product of the context-free derivation probabilities of each word w_i as (2).

$$P(W|L) = \prod_i P(w_i|l_i) \quad (2)$$

The third submodel is the lexical dependency model $D(W|R)$, which reflects word collocation statistics. It is given by the product of the lexical dependency parameter $D(w_i|l_i[c_i])$ as given in (3).

$$D(W|R) = \prod_i D(w_i|l_i[c_i]) \quad (3)$$

In (3), each c_i is the subset of W which has the strongest influence on the derivation $l_i \rightarrow w_i$. We call c_i the lexical context of w_i . The lexical dependency parameter is given by (4).

$$D(w_i|l_i[c_i]) = \frac{P(w_i|l_i[c_i])}{P(w_i|l_i)} \quad (4)$$

$D(w_i|l_i[c_i])$ measures the degree of dependency between the lexical derivation $l_i \rightarrow w_i$ and its lexical context c_i . It is close to one if w_i and c_i are highly independent. It becomes greater than one if w_i and c_i are positively correlated, whereas it becomes less

than one and close to zero if w_i and c_i are negatively correlated.

Let us show some examples of the lexical dependency parameters. Given a verb v which subordinates slot-markers s_1, \dots, s_n , the following lexical dependency parameter is considered.

$$D(v|V[s_1, \dots, s_n]) = \frac{P(v|V[s_1, \dots, s_n])}{P(v|V)} \quad (5)$$

In (5), V is the POS of v , and $P(v|V[s_1, \dots, s_n])$ is the probability of a lexical derivation $V \rightarrow v$, given that V subordinates slot-markers s_1, \dots, s_n . Both the dependency between the head verb and each of its slot-markers and the dependency between each of the slot-markers are reflected in (5).

When n is a slot-filler of slot s of a head word h , the following lexical dependency parameter is considered.

$$D(n|N[h, s]) = \frac{P(n|N[h, s])}{P(n|N)} \quad (6)$$

In (6), N is the POS of n , and $P(n|N[h, s])$ is the probability of a lexical derivation $N \rightarrow n$, given that N functions as a filler of slot s of a head word h . Thus (6) reflects the lexical dependency between a slot-filler and head word linked by a given slot-marker.

One of the advantages of our statistical language model given by (1) is the modularity of the different statistical types. Syntactic statistics, word occurrence statistics and word collocation statistics are reflected in the distinct submodels $P(R)$, $P(W|L)$ and $D(W|R)$, respectively, while maintaining the probabilistic well-foundedness of the overall model. The modularity of each statistical types enables us to obtain different statistics from different language resources. As described in the next section, we train the syntactic model $P(R)$ from plain text by way of an iterative learning process, while we train the derivation model $P(W|L)$ and the lexical dependency model $D(W|R)$ from existing language resources.

3. Constructing a Tree-Annotated Corpus

One of the simplest ways to annotate sentences with parse trees automatically is to use a statistical parser to analyse each sentence, and choose the most probable parse trees according to the statistical language model. The problem is how to train the statistical language model to reflect syntactic statistics, that is how to train the syntactic model $P(R)$ in our probabilistic language model given by (1), given that there is no tree-annotated corpus as training data. We use an iterative learning procedure to train each $P(R)$, i.e. we repeatedly derive syntactic trees for plain text inputs using a statistical parser, and iteratively train $P(R)$ from the obtained annotated trees. Furthermore, in order to improve the quality of tree-annotated corpus, we would allow for minimal human intervention, and make use of lexical statistics, the derivation model $P(W|L)$ and the lexical dependency model $D(W|R)$

in our probabilistic language model, all of which can be trained from existing language resources. In the following three subsections, we detail our method for constructing a tree-annotated corpus.

3.1. Our Method

The following iterative procedure provides our parse tree annotation method. An overview of our method is given in Figure 2. Given T as the set of sentences we would like to annotate, and $P_i(R)$ as the syntactic model at the i -th iteration:

1. We train the derivation model $P(W|L)$ and lexical dependency model $D(W|R)$ from existing language resources, such as POS-tagged corpora and tree-annotated corpora. Notice that these lexical statistics are trained independently of the syntactic model $P_i(R)$, before the following iterative procedure begins. Details of training each $P(W|L)$ and $D(W|R)$ are described in Section 3.2. and 3.3., respectively.
2. We manually annotate a set of sentences with syntactic trees I , and initialize syntactic model $P_1(R)$ based on them. The number of sentences in I should be as small as possible to avoid too much human effort.
3. Let $i = 1$, and repeats steps 4 and 5 until its a point of saturation is reached.
4. Let J be an empty set. We analyse each sentence in T and rank the associated parse trees in order of the overall probability returned by the language model $P_i(R) \cdot P(W|L) \cdot D(W|R)$, then add the top N parse trees to J . Although the parse tree annotation for each sentence in T is the most probable one, we use the top N parse trees to train the syntactic model $P_i(R)$ over.
5. Let $i = i + 1$, and newly train $P_i(R)$ over $I \cup J$.
6. Automatically annotated trees are postedited by human annotators.

As the current level of natural language processing technologies is not yet sufficient to construct a tree-annotated corpus fully automatically, we consider the human intervention in step 6 to be necessary to guarantee the quality of the annotated trees. Obviously, the more accurate the automatically annotated trees are, the less intervention on the part of human annotators is needed.

3.2. Training the Derivation Model

As described in Section 2., the derivation model $P(W|L)$ is the product of each $P(w|l)$, which is the probability of each derivation $l \rightarrow w$. When the POS tag set used in the parsing system is equal to that of an existing POS-tagged corpus, $P(w|l)$ can be trained from it by maximum likelihood estimation as in (7).

$$P(w|l) = \frac{O(w, l)}{\sum_{w \in l} O(w, l)} \quad (7)$$

In (7), $O(w, l)$ indicates the occurrence of word w with POS tag l in the training corpus. Because such a POS-tagged corpus won't always be available, however, we may be required to train each $P(w|l)$ from the POS-tagged corpus based on a different POS tag set to that of the lexicon used in the parsing system.

Suppose that L^s is the POS tag set used in the parsing system, and L^t is the POS tag set for the POS-tagged corpus. Furthermore, let L^c be a novel POS tag set, where each POS $l_s \in L^s$ is a subdivided POS of some unique $l_c \in L^c$, and each POS $l_t \in L^t$ is also a subdivided POS of l_c . For example, if L^s contains $\{singular_noun, plural_noun\}$ as noun POS tags and L^t contains $\{common_noun, proper_noun, pronoun\}$, we would define L^c as $\{noun\}$. It is possible to design L^c for any pair of L^s and L^t , if we define L^c as a set of coarse POSs, such as *noun*, *verb*, *adjective*, etc. Then, we estimate the derivation probability $P(w|l)$ as follows:

$$P(w|l_s) = \frac{O(w, l_c)}{\sum_{w \in l_s} O(w, l_c)} \quad (8)$$

In (8), l_s is the subdivided POS of l_c , and $O(w, l_c)$ is the occurrence of the word w whose POS tag is l_t (a subdivision of l_c), in the training corpus. For example, we estimate $P(w|singular_noun)$ using $O(w, noun)$, that is the occurrence of the word w whose POS tag is *common_noun*, *proper_noun* or *pronoun*.

3.3. Training the Lexical Dependency Model

As described in Section 2., the lexical dependency model $D(W|R)$ is the product of the lexical dependency parameters $D(w|l[c])$, as given in (3). The denominator in (4) is the same as the context-free derivation probability, and estimated by (8). The numerator in (4), on the other hand, can be estimated using word collocation data.

First, we consider the dependencies between slot-markers and their lexical head using the lexical dependency parameter (5). We estimate the numerator as follows:

$$P(v|V[s_1, \dots, s_n]) = \frac{O(v, V, s_1, \dots, s_n)}{\sum_{v \in V} O(v, V, s_1, \dots, s_n)} \quad (9)$$

In (9), $O(v, V, s_1, \dots, s_n)$ is the occurrence of verb v with POS V , governing slot-markers s_1, \dots, s_n . $O(v, V, s_1, \dots, s_n)$ can be obtained from a tree-annotated corpus easily. Although an existing tree-annotated corpus wouldn't always be appropriate for training the syntactic model due to differences in the grammar used in a parsing system and that underlying the corpus, we can apply it in acquiring cooccurrence data of head verbs and their slot-markers, and incorporate this into the overall statistical language model.

Next, we consider dependencies between slot-fillers and their head verb linked by a given slot-marker, using the lexical dependency parameter (6). We estimate the numerator of (6) as follows:

$$P(n|N[h, s]) = \frac{O(n, N, h, s)}{\sum_{n \in N} O(n, N, h, s)} \quad (10)$$

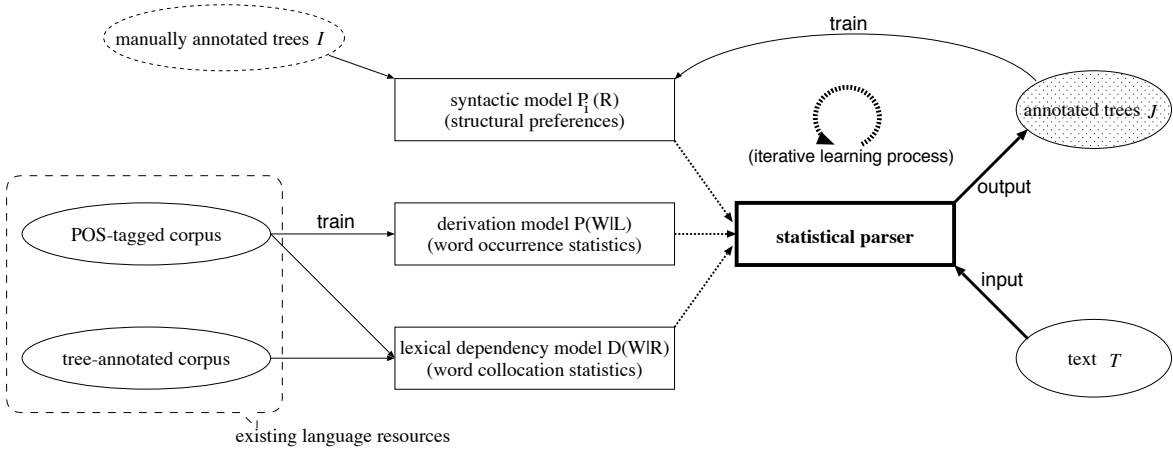


Figure 2: Overview of Constructing a Tree-Annotated Corpus

In (10), $O(n, N, h, s)$ is the occurrence of the slot-filler noun n of slot s governed by head word h , whose POS is N . This collocation data can be obtained from a POS-tagged corpus.

In the case that the POS tag set used in the parsing system is different from that of the training corpora, (9) and (10) can be estimated by way of (11) and (12), respectively, similar to the estimation of $P(w|l_s)$ in (8).

$$P(v|V_s[s_1, \dots, s_n]) = \frac{O(v, V_c, s_1, \dots, s_n)}{\sum_{v \in V_s} O(v, V_c, s_1, \dots, s_n)} \quad (11)$$

$$P(n|N_s[h, s]) = \frac{O(n, N_c, h, s)}{\sum_{n \in N_s} O(n, N_c, h, s)} \quad (12)$$

4. Experiment

By way of evaluation, we conducted an experiment to annotate plain text inputs in Japanese with syntactic trees following the method described in Section 3.1.

4.1. Our Parsing System

First, let us briefly describe our statistical parsing system. Our system is based on the MSLR parser,¹ which integrates morphological and syntactic analysis of unsegmented languages such as Japanese. The grammar consists of 1,498 context-free rules containing 220 nonterminal symbols and 556 terminal symbols (i.e. POS tags). According to this grammar, the parser generates syntactic parse trees representing bunsetu phrase (BP, hereafter) boundaries, and dependency relations between them. A BP is a chunk of words consisting of a content word (noun, verb, adjective, etc.) accompanied by some function word(s) (postposition, auxiliary, etc.). For example, the BP “*kanajo-ga*” (BP_1) in Figure 1 consists of the noun “*kanajo*” (she) followed by the postposition “*ga*” (NOM), which functions as a nominative slot-marker. The BP “*ake-ta*” (BP_3), on the other hand, consists of the verb “*ake*” (open) followed by the auxiliary “*ta*” (PAST). In

Japanese, when BP_i precedes BP_j and BP_i and BP_j are in a dependency relation, BP_i is always the modifier of BP_j . For example, in Figure 1, both BP_1 and BP_2 modify BP_3 . Furthermore, our parser interfaces with the EDR Japanese dictionary (EDR, 1995) consisting of 241,189 words. Using the grammar and dictionary described above, the parser analyzes sentences, generates parse trees, and ranks them in order of overall probability, i.e. the product of the three submodels $P(R) \cdot P(W|L) \cdot D(W|R)$.

4.2. Training the Statistical Language Model and Generating Trees

Of the three submodels, the derivation model $P(W|L)$ and lexical dependency model $D(W|R)$ are trained from existing language resources independently of the iterative annotation process. For training the derivation model $P(W|L)$, we extracted about 123 million words and their associated POS tags from the RWC POS-tagged corpus (Hasida et al., 1998). As the POS tag set for the RWC corpus differs from that for our parsing system, we designed L^c consisting of 16 coarse POSs and estimated $P(w|l_s)$ by (8).

In order to train lexical dependency parameter (5), we extracted about 400,000 collocation instances of a verb v and its subordinating slot-markers s_1, \dots, s_n from the EDR Japanese corpus (EDR, 1995), where a skeleton tree without labels on intermediate nodes is provided for each sentence. Notice that the EDR corpus is not appropriate to train the syntactic model $P(R)$ on, because the grammar underlying the trees in the EDR corpus is quite different from that of our parsing system. On the other hand, lexical dependency parameter (6) was trained using 6.7 million instances of (noun, verb, slot-marker) collocation collected from both the EDR corpus and the RWC corpus. Similar to the case of learning each $P(w|l)$, we used the set of coarse POSs L^c and estimated the numerator of dependency parameters (5) and (6) by (11) and (12), respectively.

Then, we went through 5 iterations of annotating plain text input with syntactic trees and training $P_i(R)$

¹<http://tanaka-www.cs.titech.ac.jp/pub/mslr/>

Table 1: Result of Experiment 1

i	$ I = 0$			
	BPD	BPB	WDP	WDS
1	8 %	27 %	30 %	52 %
2	8 %	27 %	31 %	52 %
3	8 %	27 %	31 %	51 %
4	8 %	28 %	31 %	51 %
5	8 %	28 %	31 %	51 %
i	$ I = 50$			
	BPD	BPB	WDP	WDS
1	8 %	28 %	32 %	52 %
2	7 %	24 %	27 %	51 %
3	8 %	24 %	27 %	50 %
4	9 %	25 %	28 %	50 %
5	9 %	24 %	26 %	50 %
i	$ I = 100$			
	BPD	BPB	WDP	WDS
1	9 %	25 %	31 %	52 %
2	8 %	23 %	26 %	50 %
3	8 %	24 %	27 %	50 %
4	9 %	26 %	29 %	50 %
5	9 %	25 %	27 %	50 %

from the obtained annotated trees, as described in Section 3.1. As set T , we used 10,000 sentences extracted from the EDR Japanese corpus. In this experiment, we set $N = 10$, that is we used the top N parse trees for each sentence to train the syntactic model $P_i(R)$ over. Furthermore, we variously set $|I|$, the number of manually annotated trees used for training the initial syntactic model $P_1(R)$, to 0, 50 and 100. When $|I| = 0$, the initial syntactic model $P_1(R)$ was trained as follows: we analysed sentences in T , chose a parse tree semi-randomly for each sentence and trained $P_1(R)$ according to these parse trees. In choosing a parse tree, we preferred trees containing the least number of words from among the various word segmentation candidates, and also trees containing the least number of BPs.

4.3. Results

We selected 100 sentences from T and evaluated the syntactic trees provided for them at each iteration i . The results are shown in Tables 1 and 2.

WDS, WDP, BPB and BPD in Table 1 indicate sentence accuracies defined as follows:

WDS

of sentences where word segmentation was correct

total number of sentences

WDP

of sentences where word segmentation and POS tagging were correct

total number of sentences

BPB

Table 2: Result of Experiment 2

i	$ I = 0$		
	R-MOR	P-MOR	A-BP
1	91.37 %	93.60 %	77.78 %
2	92.15 %	93.77 %	76.65 %
3	92.32 %	93.79 %	77.84 %
4	92.49 %	93.96 %	76.89 %
5	92.49 %	93.96 %	76.30 %
i	$ I = 50$		
	R-MOR	P-MOR	A-BP
1	91.37 %	93.56 %	77.33 %
2	91.80 %	93.46 %	77.55 %
3	91.58 %	93.28 %	79.72 %
4	91.84 %	93.50 %	81.46 %
5	91.84 %	93.34 %	81.69 %
i	$ I = 100$		
	R-MOR	P-MOR	A-BP
1	91.28 %	93.55 %	80.41 %
2	91.58 %	93.16 %	80.29 %
3	91.67 %	93.33 %	79.02 %
4	91.93 %	93.51 %	81.25 %
5	91.80 %	93.30 %	80.79 %

of sentences where BP boundaries were correct, in addition to correct word segmentation and POS tagging

total number of sentences

BPD

of sentences where BP boundaries and inter-BP dependencies were correct, in addition to correct word segmentation and POS tagging

total number of sentences

R-MOR, P-MOR and A-BP in table 2 indicates the accuracies defined as follows:

R-MOR Recall of morphological analysis.

of words correctly segmented and POS tagged

of words in the solution set

P-MOR Precision of morphological analysis.

of words correctly segmented and POS tagged

of words in the automatically annotated trees

A-BP

of BPs whose modifier was correctly identified

total number of BPs

Notice that the figures for A-BP pertain only to those sentences where the BP boundaries were correctly identified.

As shown in Table 2, the more syntactic trees are manually annotated beforehand, the greater the A-BP accuracy in automatic tree annotation is, even when the number of manually annotated sentences is very small. On the other hand, the gain in the sentence accuracy realized by iterative learning of the syntactic model is not that great. There is thus much room for improvement in our iterative learning method.

5. Conclusion

In this paper, we proposed a method to construct a tree-annotated corpus using an iterative learning statistical language model. We allowed for human intervention in two ways, first is providing the syntactic tree for a small amount of sentences in order to initialize the syntactic model, and second is postediting the automatically generated syntactic trees. We also trained the lexical statistics, such as word occurrence statistics and word collocation statistics, from existing language resources, and used them in the iterative annotation procedure.

In the future, we hope to examine reasons why our iterative learning procedure didn't work as well as expected, and find ways to improve it. In addition to this, we plan to develop tools which can display the syntactic trees graphically and facilitate their easy modification to reduce the burden on human annotators in postediting the annotated trees.

6. Acknowledgement

We would like to thank the reviewers for their suggestive comments. We would also like to thank Timothy Baldwin (Tokyo Institute of Technology) for his help in writing this paper.

7. References

- Charniak, Eugene, 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the National Conference on Artificial Intelligence*.
- Collins, Michael, 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- EDR, 1995. The EDR electronic dictionary technical guide (second edition). Technical Report TR-045, Japan Electronic Dictionary Research Institute.
- Fujio, Masakazu and Yuji Matsumoto, 1998. Japanese dependency structure analysis based on lexicalized statistics. In *Proceedings of the the 3rd Conference on Empirical Methods in Natural Language Processing*.
- Hasida, Koiti, Hitoshi Isahara, Takenobu Tokunaga, Minako Hashimoto, Shiho Ogino, Wakako Kashino, Jun Toyoura, and Hironobu Takahashi, 1998. The RWC text databases. In *Proceedings of the the First International Conference on Language Resources and Evaluation*.
- Inui, Kentaro, Virach Sornlertlamvanich, Hozumi Tanaka, and Takenobu Tokunaga, 1997. A new formalization of probabilistic GLR parsing. In *Proceedings of the IWPT*.
- Inui, Kouji and Kentaro Inui, 2000. Committee-based probabilistic partial parsing. In *Proceedings of the Annual Meeting of the Japan Association for Natural Language Processing*. (in Japanese).
- Kurohashi, Sadao and Makoto Nagao, 1997. Kyoto University text corpus project. In *Proceedings of the 11th Annual Conference of JSAI*. (In Japanese).
- Lari, K. and S. J. Young, 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Languages*, 4:35–56.
- Li, Hang, 1996. A probabilistic disambiguation method based on psycholinguistic principles. In *Proceedings of the Workshop on Very Large Corpora*.
- Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz, 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Shirai, Kiyooki, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka, 1998. An empirical evaluation on statistical parsing of Japanese sentences using lexical association statistics. In *Proceedings of the the 3rd Conference on Empirical Methods in Natural Language Processing*.
- Sornlertlamvanich, Virach, Kentaro Inui, Kiyooki Shirai, Hozumi Tanaka, Takenobu Tokunaga, and Toshiyuki Takezawa, 1997. Empirical evaluation of probabilistic GLR parsing. In *Proceedings of the Natural Language Processing Pacific Rim Symposium*.
- Uchimoto, Kiyotaka, Satoshi Sekine, and Hitoshi Isahara, 1999. Japanese dependency structure analysis based on maximum entropy models. *Journal of the Information Processing Society of Japan*, 40(9):3397–3407. (in Japanese).