A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis

Daisuke Kawahara[*] and **Sadao Kurohashi**[†]
Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo Bunkyo-ku, Tokyo, 113-8656, Japan
{kawahara,kuro}@kc.t.u-tokyo.ac.jp

Abstract

We present an integrated probabilistic model for Japanese syntactic and case structure analysis. Syntactic and case structure are simultaneously analyzed based on wide-coverage case frames that are constructed from a huge raw corpus in an unsupervised manner. This model selects the syntactic and case structure that has the highest generative probability. We evaluate both syntactic structure and case structure. In particular, the experimental results for syntactic analysis on web sentences show that the proposed model significantly outperforms known syntactic analyzers.

1 Introduction

Case structure (predicate-argument structure or logical form) represents what arguments are related to a predicate, and forms a basic unit for conveying the meaning of natural language text. Identifying such case structure plays an important role in natural language understanding.

In English, syntactic case structure can be mostly derived from word order. For example, the left argument of the predicate is the subject, and the right argument of the predicate is the object in most cases. Blaheta and Charniak proposed a statistical method for analyzing function tags in Penn Treebank, and achieved a really high accuracy of 95.7% for syntactic roles, such as SBJ (subject) and DTV (dative) (Blaheta and Charniak, 2000). In recent years, there have been many studies on semantic structure analysis (semantic role labeling) based on PropBank (Kingsbury et al., 2002) and FrameNet (Baker et al., 1998). These studies classify syntactic roles into semantic ones such as agent, experiencer and instrument.

Case structure analysis of Japanese is very different from that of English. In Japanese, postpositions are used to mark cases. Frequently used postpositions are "*ga*", "*wo*" and "*ni*", which usually mean nominative, accusative and dative. However, when an argument is followed by the topic-marking postposition "*wa*", its case marker is hidden. In addition, case-marking postpositions are often omitted in Japanese. These troublesome characteristics make Japanese case structure analysis very difficult.

To address these problems and realize Japanese case structure analysis, wide-coverage case frames are required. For example, let us describe how to apply case structure analysis to the following sentence:

bentou-wa	*taberu*
lunchbox-TM	eat
(eat lunchbox)	

In this sentence, *taberu* (eat) is a verb, and *bentou-wa* (lunchbox-TM) is a case component (i.e. argument) of *taberu*. The case marker of "*bentou-wa*" is hidden by the topic marker (TM) "*wa*". The analyzer matches "*bentou*" (lunchbox) with the most

[*] Currently, National Institute of Information and Communications Technology, JAPAN, dk@nict.go.jp

[†] Currently, Graduate School of Informatics, Kyoto University, kuro@i.kyoto-u.ac.jp

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 176–183,
New York, June 2006.

suitable case slot (CS) in the following case frame of "*taberu*" (eat).

	CS	examples
taberu	*ga*	person, child, boy, $\cdots$
	wo	lunch, lunchbox, dinner, $\cdots$

Since "*bentou*" (lunchbox) is included in "*wo*" examples, its case is analyzed as "*wo*". As a result, we obtain the case structure "ϕ:*ga bentou*:*wo taberu*", which means that "*ga*" (nominative) argument is omitted, and "*wo*" (accusative) argument is "*bentou*" (lunchbox). In this paper, we run such case structure analysis based on example-based case frames that are constructed from a huge raw corpus in an unsupervised manner.

Let us consider syntactic analysis, into which our method of case structure analysis is integrated. Recently, many accurate statistical parsers have been proposed (e.g., (Collins, 1999; Charniak, 2000) for English, (Uchimoto et al., 2000; Kudo and Matsumoto, 2002) for Japanese). Since they somehow use lexical information in the tagged corpus, they are called "lexicalized parsers". On the other hand, unlexicalized parsers achieved an almost equivalent accuracy to such lexicalized parsers (Klein and Manning, 2003; Kurohashi and Nagao, 1994). Accordingly, we can say that the state-of-the-art lexicalized parsers are mainly based on unlexical (grammatical) information due to the sparse data problem. Bikel also indicated that Collins' parser can use bilexical dependencies only 1.49% of the time; the rest of the time, it backs off to condition one word on just phrasal and part-of-speech categories (Bikel, 2004).

This paper aims at exploiting much more lexical information, and proposes a fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. Lexical information is extracted not from a small tagged corpus, but from a huge raw corpus as case frames. This model performs case structure analysis by a generative probabilistic model based on the case frames, and selects the syntactic structure that has the highest case structure probability.

2 Automatically Constructed Case Frames

We employ automatically constructed case frames (Kawahara and Kurohashi, 2002) for our model of case structure analysis. This section outlines the method for constructing the case frames.

Table 1: Case frame examples (examples are expressed only in English for space limitation.).

	CS	examples
youritsu (1) (support)	*ga*	<agent>, group, party, $\cdots$
	wo	<agent>, candidate, applicant
	ni	<agent>, district, election, $\cdots$
youritsu (2) (support)	*ga*	<agent>
	wo	<agent>, member, minister, $\cdots$
	ni	<agent>, candidate, successor
$\vdots$	$\vdots$	$\vdots$
itadaku (1) (have)	*ga*	<agent>
	wo	soup
itadaku (2) (be given)	*ga*	<agent>
	wo	advice, instruction, address
	kara	<agent>, president, circle, $\cdots$
$\vdots$	$\vdots$	$\vdots$

A large corpus is automatically parsed, and case frames are constructed from modifier-head examples in the resulting parses. The problems of automatic case frame construction are syntactic and semantic ambiguities. That is to say, the parsing results inevitably contain errors, and verb senses are intrinsically ambiguous. To cope with these problems, case frames are gradually constructed from reliable modifier-head examples.

First, modifier-head examples that have no syntactic ambiguity are extracted, and they are disambiguated by a couple of a verb and its closest case component. Such couples are explicitly expressed on the surface of text, and can be considered to play an important role in sentence meanings. For instance, examples are distinguished not by verbs (e.g., "*tsumu*" (load/accumulate)), but by couples (e.g., "*nimotsu-wo tsumu*" (load baggage) and "*keiken-wo tsumu*" (accumulate experience)). Modifier-head examples are aggregated in this way, and yield basic case frames.

Thereafter, the basic case frames are clustered to merge similar case frames. For example, since "*nimotsu-wo tsumu*" (load baggage) and "*busshi-wo tsumu*" (load supply) are similar, they are clustered. The similarity is measured using a thesaurus (Ikehara et al., 1997).

Using this gradual procedure, we constructed case frames from the web corpus (Kawahara and Kuro-

hashi, 2006). The case frames were obtained from approximately 470M sentences extracted from the web. They consisted of 90,000 verbs, and the average number of case frames for a verb was 34.3.

In Figure 1, some examples of the resulting case frames are shown. In this table, 'CS' means a case slot. <agent> in the table is a generalized example, which is given to the case slot where half of the examples belong to <agent> in a thesaurus (Ikehara et al., 1997). <agent> is also given to "*ga*" case slot that has no examples, because "*ga*" case components are usually agentive and often omitted.

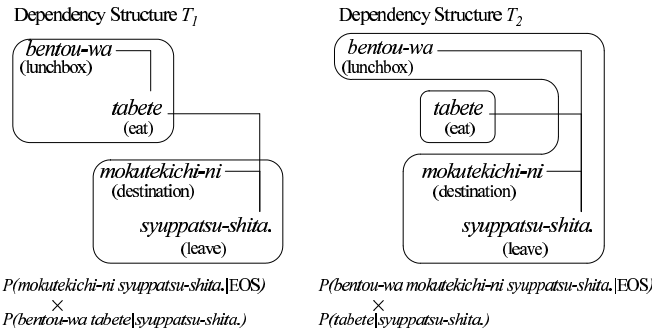

Figure 1: An Example of Probability Calculation.

3 Integrated Probabilistic Model for Syntactic and Case Structure Analysis

The proposed method gives a probability to each possible syntactic structure T and case structure L of the input sentence S, and outputs the syntactic and case structure that have the highest probability. That is to say, the system selects the syntactic structure T_{best} and the case structure L_{best} that maximize the probability $P(T, L|S)$:

$$\begin{aligned}(T_{best}, L_{best}) &= \underset{(T,L)}{\operatorname{argmax}} P(T, L|S) \\ &= \underset{(T,L)}{\operatorname{argmax}} \frac{P(T, L, S)}{P(S)} \\ &= \underset{(T,L)}{\operatorname{argmax}} P(T, L, S) \qquad (1)\end{aligned}$$

The last equation is derived because $P(S)$ is constant.

3.1 Generative Model for Syntactic and Case Structure Analysis

We propose a generative probabilistic model based on the dependency formalism. This model considers a clause as a unit of generation, and generates the input sentence from the end of the sentence in turn. $P(T, L, S)$ is defined as the product of a probability for generating a clause C_i as follows:

$$P(T, L, S) = \prod_{i=1..n} P(C_i|b_{h_i}) \qquad (2)$$

where n is the number of clauses in S, and b_{h_i} is C_i's modifying *bunsetsu*[1]. The main clause C_n at the end of a sentence does not have a modifying head, but we handle it by assuming b_{h_n} = EOS (End Of Sentence).

For example, consider the sentence in Figure 1. There are two possible dependency structures, and for each structure the product of probabilities indicated below of the tree is calculated. Finally, the model chooses the highest-probability structure (in this case the left one).

C_i is decomposed into its predicate type f_i (including the predicate's inflection) and the rest case structure CS_i. This means that the predicate included in CS_i is lemmatized. *Bunsetsu* b_{h_i} is also decomposed into the content part w_{h_i} and the type f_{h_i}.

$$\begin{aligned}P(C_i|b_{h_i}) &= P(CS_i, f_i|w_{h_i}, f_{h_i}) \\ &= P(CS_i|f_i, w_{h_i}, f_{h_i})P(f_i|w_{h_i}, f_{h_i}) \\ &\approx P(CS_i|f_i, w_{h_i})P(f_i|f_{h_i}) \qquad (3)\end{aligned}$$

The last equation is derived because the content part in CS_i is independent of the type of its modifying head (f_{h_i}), and in most cases, the type f_i is independent of the content part of its modifying head (w_{h_i}).

For example, $P(\textit{bentou-wa tabete}|\textit{syuppatsu-shita})$ is calculated as follows:

$$P(CS(\textit{bentou-wa taberu})|\textit{te}, \textit{syuppatsu-suru})P(\textit{te}|\textit{ta.})$$

We call $P(CS_i|f_i, w_{h_i})$ *generative model for case structure* and $P(f_i|f_{h_i})$ *generative model for predicate type*. The following two sections describe these models.

3.2 Generative Model for Case Structure

We propose a generative probabilistic model of case structure. This model selects a case frame that

[1] In Japanese, *bunsetsu* is a basic unit of dependency, consisting of one or more content words and the following zero or more function words. It corresponds to a base phrase in English, and "*eojeol*" in Korean.

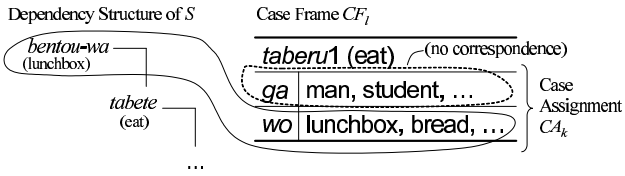

Figure 2: An example of case assignment CA_k.

matches the input case components, and makes correspondences between input case components and case slots.

A case structure CS_i consists of a predicate v_i, a case frame CF_l and a case assignment CA_k. Case assignment CA_k represents correspondences between input case components and case slots as shown in Figure 2. Note that there are various possibilities of case assignment in addition to that of Figure 2, such as corresponding "*bentou*" (lunchbox) with "*ga*" case. Accordingly, the index k of CA_k ranges up to the number of possible case assignments. By splitting CS_i into v_i, CF_l and CA_k, $P(CS_i|f_i, w_{h_i})$ is rewritten as follows:

$$\begin{aligned} P(CS_i|f_i, w_{h_i}) &= P(v_i, CF_l, CA_k|f_i, w_{h_i}) \\ &= P(v_i|f_i, w_{h_i}) \\ &\quad \times P(CF_l|f_i, w_{h_i}, v_i) \\ &\quad \times P(CA_k|f_i, w_{h_i}, v_i, CF_l) \\ &\approx P(v_i|w_{h_i}) \\ &\quad \times P(CF_l|v_i) \\ &\quad \times P(CA_k|CF_l, f_i) \end{aligned} \quad (4)$$

The above approximation is given because it is natural to consider that the predicate v_i depends on its modifying head w_{h_i}, that the case frame CF_l only depends on the predicate v_i, and that the case assignment CA_k depends on the case frame CF_l and the predicate type f_i.

The probabilities $P(v_i|w_{h_i})$ and $P(CF_l|v_i)$ are estimated from case structure analysis results of a large raw corpus. The remainder of this section illustrates $P(CA_k|CF_l, f_i)$ in detail.

3.2.1 Generative Probability of Case Assignment

Let us consider case assignment CA_k for each case slot s_j in case frame CF_l. $P(CA_k|CF_l, f_i)$ can be decomposed into the following product depending on whether a case slot s_j is filled with an input case component (content part n_j and type f_j) or vacant:

$$\begin{aligned} &P(CA_k|CF_l, f_i) = \\ &\quad \prod_{s_j:A(s_j)=1} P(A(s_j)=1, n_j, f_j|CF_l, f_i, s_j) \\ &\times \prod_{s_j:A(s_j)=0} P(A(s_j)=0|CF_l, f_i, s_j) \\ &= \prod_{s_j:A(s_j)=1} \Big\{ P(A(s_j)=1|CF_l, f_i, s_j) \\ &\qquad \times P(n_j, f_j|CF_l, f_i, A(s_j)=1, s_j) \Big\} \\ &\times \prod_{s_j:A(s_j)=0} P(A(s_j)=0|CF_l, f_i, s_j) \end{aligned} \quad (5)$$

where the function $A(s_j)$ returns 1 if a case slot s_j is filled with an input case component; otherwise 0.

$P(A(s_j) = 1|CF_l, f_i, s_j)$ and $P(A(s_j) = 0|CF_l, f_i, s_j)$ in equation (5) can be rewritten as $P(A(s_j) = 1|CF_l, s_j)$ and $P(A(s_j) = 0|CF_l, s_j)$, because the evaluation of case slot assignment depends only on the case frame. We call these probabilities *generative probability of a case slot*, and they are estimated from case structure analysis results of a large corpus.

Let us calculate $P(CS_i|f_i, w_{h_i})$ using the example in Figure 1. In the sentence, "*wa*" is a topic marking (TM) postposition, and hides the case marker. The generative probability of case structure varies depending on the case slot to which the topic marked phrase is assigned. For example, when a case frame of "*taberu*" (eat) $CF_{taberu1}$ with "*ga*" and "*wo*" case slots is used, $P(CS(\textit{bentou-wa taberu})|\textit{te}, \textit{syuppatsu-suru})$ is calculated as follows:

$$\begin{aligned} &P_1(CS(\textit{bentou-wa taberu})|\textit{te}, \textit{syuppatsu-suru}) = \\ &\quad P(\textit{taberu}|\textit{syuppatsu-suru}) \\ &\times \; P(CF_{\textit{taberu1}}|\textit{taberu}) \\ &\times \; P(\textit{bentou}, \textit{wa}|CF_{\textit{taberu1}}, \textit{te}, A(\textit{wo})=1, \textit{wo}) \\ &\times \; P(A(\textit{wo})=1|CF_{\textit{taberu1}}, \textit{wo}) \\ &\times \; P(A(\textit{ga})=0|CF_{\textit{taberu1}}, \textit{ga}) \end{aligned} \quad (6)$$

$$
\begin{aligned}
&P_2(CS(\textit{bentou-wa taberu})|\textit{te}, \textit{syupatsu-suru}) = \\
&\quad P(\textit{taberu}|\textit{syuppatsu-suru}) \\
&\times \quad P(CF_{\textit{taberu1}}|\textit{taberu}) \\
&\times \quad P(\textit{bentou}, \textit{wa}|CF_{\textit{taberu1}}, \textit{te}, A(\textit{ga}) = 1, \textit{ga}) \\
&\times \quad P(A(\textit{ga}) = 1|CF_{\textit{taberu1}}, \textit{ga}) \\
&\times \quad P(A(\textit{wo}) = 0|CF_{\textit{taberu1}}, \textit{wo}) \qquad (7)
\end{aligned}
$$

Such probabilities are computed for each case frame of "*taberu*" (eat), and the case frame and its corresponding case assignment that have the highest probability are selected.

We describe the generative probability of a case component $P(n_j, f_j|CF_l, f_i, A(s_j) = 1, s_j)$ below.

3.2.2 Generative Probability of Case Component

We approximate the generative probability of a case component, assuming that:

- a generative probability of content part n_j is independent of that of type f_j,
- and the interpretation of the surface case included in f_j does not depend on case frames.

Taking into account these assumptions, the generative probability of a case component is approximated as follows:

$$
\begin{aligned}
&P(n_j, f_j|CF_l, f_i, A(s_j) = 1, s_j) \approx \\
&\quad P(n_j|CF_l, A(s_j) = 1, s_j)\ P(f_j|s_j, f_i) \qquad (8)
\end{aligned}
$$

$P(n_j|CF_l, A(s_j) = 1, s_j)$ is the probability of generating a content part n_j from a case slot s_j in a case frame CF_l. This probability is estimated from case frames.

Let us consider $P(f_j|s_j, f_i)$ in equation (8). This is the probability of generating the type f_j of a case component that has a correspondence with the case slot s_j. Since the type f_j consists of a surface case c_j[2], a punctuation mark (comma) p_j and a topic marker "*wa*" t_j, $P(f_j|s_j, f_i)$ is rewritten as follows (using the chain rule):

$$
\begin{aligned}
P(f_j|s_j, f_i) &= P(c_j, t_j, p_j|s_j, f_i) \\
&= P(c_j|s_j, f_i) \\
&\quad \times P(p_j|s_j, f_i, c_j) \\
&\quad \times P(t_j|s_j, f_i, c_j, p_j) \\
&\approx P(c_j|s_j) \\
&\quad \times P(p_j|f_i) \\
&\quad \times P(t_j|f_i, p_j) \qquad (9)
\end{aligned}
$$

This approximation is given by assuming that c_j only depends on s_j, p_j only depends on f_j, and t_j depends on f_j and p_j. $P(c_j|s_j)$ is estimated from the Kyoto Text Corpus (Kawahara et al., 2002), in which the relationship between a surface case marker and a case slot is annotated by hand.

In Japanese, a punctuation mark and a topic marker are likely to be used when their belonging *bunsetsu* has a long distance dependency. By considering such tendency, f_i can be regarded as (o_i, u_i), where o_i means whether a dependent *bunsetsu* gets over another head candidate before its modifying head v_i, and u_i means a clause type of v_i. The value of o_i is binary, and u_i is one of the clause types described in (Kawahara and Kurohashi, 1999).

$$P(p_j|f_i) = P(p_j|o_i, u_i) \qquad (10)$$

$$P(t_j|f_i, p_j) = P(t_j|o_i, u_i, p_j) \qquad (11)$$

3.3 Generative Model for Predicate Type

Now, consider $P(f_i|f_{h_i})$ in the equation (3). This is the probability of generating the predicate type of a clause C_i that modifies b_{h_i}. This probability varies depending on the type of b_{h_i}.

When b_{h_i} is a predicate *bunsetsu*, C_i is a subordinate clause embedded in the clause of b_{h_i}. As for the types f_i and f_{h_i}, it is necessary to consider punctuation marks (p_i, p_{h_i}) and clause types (u_i, u_{h_i}). To capture a long distance dependency indicated by punctuation marks, o_{h_i} (whether C_i has a possible head candidate before b_{h_i}) is also considered.

$$P_{VBmod}(f_i|f_{h_i}) = P_{VBmod}(p_i, u_i|p_{h_i}, u_{h_i}, o_{h_i}) \qquad (12)$$

When b_{h_i} is a noun *bunsetsu*, C_i is an embedded clause in b_{h_i}. In this case, clause types and a punctuation mark of the modifiee do not affect the probability.

$$P_{NBmod}(f_i|f_{h_i}) = P_{NBmod}(p_i|o_{h_i}) \qquad (13)$$

[2] A surface case means a postposition sequence at the end of *bunsetsu*, such as "*ga*", "*wo*", "*koso*" and "*demo*".

Table 2: Data for parameter estimation.

probability	what is generated	data
$P(p_j\|o_i, u_j)$	punctuation mark	Kyoto Text Corpus
$P(t_j\|o_i, u_i, p_j)$	topic marker	Kyoto Text Corpus
$P(p_i, u_i\|p_{h_i}, u_{h_i}, o_{h_i})$	predicate type	Kyoto Text Corpus
$P(c_j\|s_j)$	surface case	Kyoto Text Corpus
$P(v_i\|w_{h_i})$	predicate	parsing results
$P(n_j\|CF_l, A(s_j) = 1, s_j)$	words	case frames
$P(CF_l\|v_i)$	case frame	case structure analysis results
$P(A(s_j) = \{0, 1\}\|CF_l, s_j)$	case slot	case structure analysis results

Table 3: Experimental results for syntactic analysis.

	baseline	proposed
all	3,447/3,976 (86.7%)	3,477/3,976 (87.4%)
NB→VB	1,310/1,547 (84.7%)	1,328/1,547 (85.8%)
TM	244/298 (81.9%)	242/298 (81.2%)
others	1,066/1,249 (85.3%)	1,086/1,249 (86.9%)
NB→NB	525/556 (94.4%)	526/556 (94.6%)
VB→VB	593/760 (78.0%)	601/760 (79.1%)
VB→NB	453/497 (91.1%)	457/497 (92.0%)

4 Experiments

We evaluated the syntactic structure and case structure outputted by our model. Each parameter is estimated using maximum likelihood from the data described in Table 2. All of these data are not existing or obtainable by a single process, but acquired by applying syntactic analysis, case frame construction and case structure analysis in turn. The process of case structure analysis in this table is a similarity-based method (Kawahara and Kurohashi, 2002). The case frames were automatically constructed from the web corpus comprising 470M sentences, and the case structure analysis results were obtained from 6M sentences in the web corpus.

The rest of this section first describes the experiments for syntactic structure, and then reports the experiments for case structure.

4.1 Experiments for Syntactic Structure

We evaluated syntactic structures analyzed by the proposed model. Our experiments were run on hand-annotated 675 web sentences [3]. The web sentences were manually annotated using the same criteria as the Kyoto Text Corpus. The system input was tagged automatically using the JUMAN morphological analyzer (Kurohashi et al., 1994). The syntactic structures obtained were evaluated with regard to dependency accuracy — the proportion of correct dependencies out of all dependencies except for the last dependency in the sentence end [4].

Table 3 shows the dependency accuracy. In the table, "baseline" means the rule-based syntactic parser, KNP (Kurohashi and Nagao, 1994), and "proposed" represents the proposed method. The proposed method significantly outperformed the baseline method (McNemar's test; $p < 0.05$). The dependency accuracies are classified into four types according to the *bunsetsu* classes (VB: verb *bunsetsu*, NB: noun *bunsetsu*) of a dependent and its head. The "NB→VB" type is further divided into two types: "TM" and "others". The type that is most related to case structure is "others" in "NB→VB". Its accuracy was improved by 1.6%, and the error rate was reduced by 10.9%. This result indicated that the proposed method is effective in analyzing dependencies related to case structure.

Figure 3 shows some analysis results, where the dotted lines represent the analysis by the baseline method, and the solid lines represent the analysis by the proposed method. Sentence (1) and (2) are incorrectly analyzed by the baseline but correctly analyzed by the proposed method.

There are two major causes that led to analysis errors.

Mismatch between analysis results and annotation criteria

In sentence (3) in Figure 3, the baseline method correctly recognized the head of "*iin-wa*" (commissioner-TM) as "*hirakimasu*" (open). However, the proposed method incorrectly judged it as "*oujite-imasuga*" (offer). Both analysis results can be considered to be correct semantically, but from

[3] The test set is not used for case frame construction and probability estimation.

[4] Since Japanese is head-final, the second last *bunsetsu* unambiguously depends on the last *bunsetsu*, and the last *bunsetsu* has no dependency.

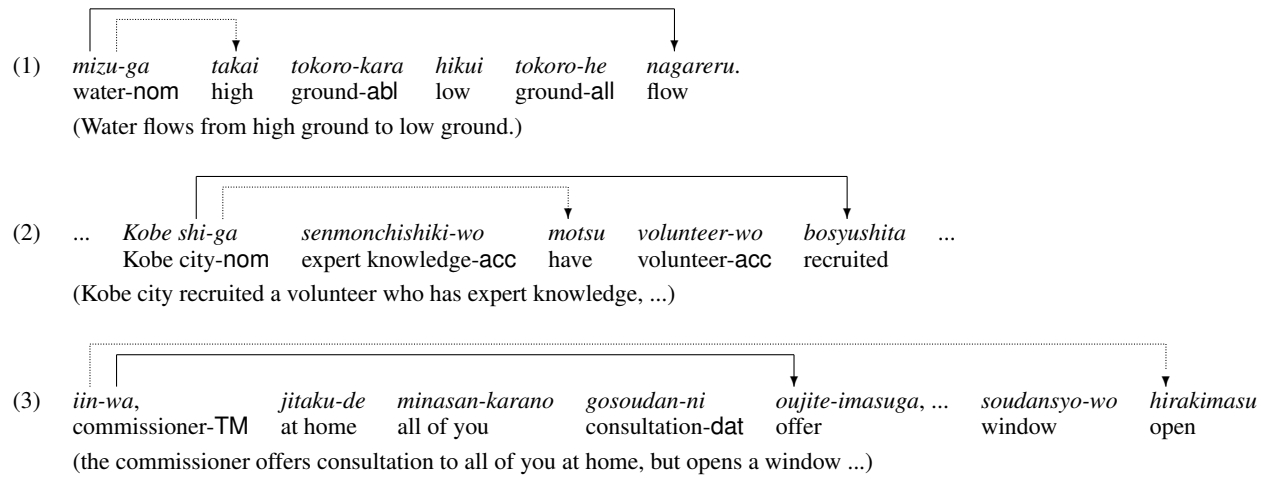

Figure 3: Examples of analysis results.

Table 4: Experimental results for case structure analysis.

	baseline	proposed
TM	72/105 (68.6%)	82/105 (78.1%)
clause	107/155 (69.0%)	121/155 (78.1%)

the viewpoint of our annotation criteria, the latter is not a syntactic relation, but an ellipsis relation. To address this problem, it is necessary to simultaneously evaluate not only syntactic relations but also indirect relations, such as ellipses and anaphora.

Linear weighting on each probability

We proposed a generative probabilistic model, and thus cannot optimize the weight of each probability. Such optimization could be a way to improve the system performance. In the future, we plan to employ a machine learning technique for the optimization.

4.2 Experiments for Case Structure

We applied case structure analysis to 215 web sentences which are manually annotated with case structure, and evaluated case markers of TM phrases and clausal modifiees by comparing them with the gold standard in the corpus. The experimental results are shown in table 4, in which the baseline refers to a similarity-based method (Kawahara and Kurohashi, 2002). The experimental results were really good compared to the baseline. It is difficult to compare the results with the previous work stated in the next section, because of different experimental settings (e.g., our evaluation includes parse errors in incorrect cases).

5 Related Work

There have been several approaches for syntactic analysis handling lexical preference on a large scale. Shirai et al. proposed a PGLR-based syntactic analysis method using large-scale lexical preference (Shirai et al., 1998). Their system learned lexical preference from a large newspaper corpus (articles of five years), such as $P(\text{pie}|wo, taberu)$, but did not deal with verb sense ambiguity. They reported 84.34% accuracy on 500 relatively short sentences from the Kyoto Text Corpus.

Fujio and Matsumoto presented a syntactic analysis method based on lexical statistics (Fujio and Matsumoto, 1998). They made use of a probabilistic model defined by the product of a probability of having a dependency between two cooccurring words and a distance probability. The model was trained on the EDR corpus, and performed with 86.89% accuracy on 10,000 sentences from the EDR corpus [5].

On the other hand, there have been a number of machine learning-based approaches using lexical preference as their features. Among these, Kudo and Matsumoto yielded the best performance (Kudo and Matsumoto, 2002). They proposed a chunking-based dependency analysis method using Support Vector Machines. They used two-fold cross validation on the Kyoto Text Corpus, and achieved 90.46%

[5]The evaluation includes the last dependencies in the sentence end, which are always correct.

accuracy [5]. However, it is very hard to learn sufficient lexical preference from several tens of thousands sentences of a hand-tagged corpus.

There has been some related work analyzing clausal modifiees and TM phrases. For example, Torisawa analyzed TM phrases using predicate-argument cooccurences and word classifications induced by the EM algorithm (Torisawa, 2001). Its accuracy was approximately 88% for "*wa*" and 84% for "*mo*". It is difficult to compare the accuracy of their system to ours, because the range of target expressions is different. Unlike related work, it is promising to utilize the resultant case frames for subsequent analyzes such as ellipsis or discourse analysis.

6 Conclusion

We have described an integrated probabilistic model for syntactic and case structure analysis. This model takes advantage of lexical selectional preference of large-scale case frames, and performs syntactic and case analysis simultaneously. The experiments indicated the effectiveness of our model. In the future, by incorporating ellipsis resolution, we will develop an integrated model of syntactic, case and ellipsis analysis.

References

Collin Baker, Charles Fillmore, and John Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, pages 86–90.

Daniel M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Don Blaheta and Eugene Charniak. 2000. Assigning function tags to parsed text. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 234–240.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Masakazu Fujio and Yuji Matsumoto. 1998. Japanese dependency structure analysis based on lexicalized statistics. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, pages 88–96.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentarou Ogura, Yoshifumi Oyama, and Yoshihiko Hayashi, editors. 1997. *Japanese Lexicon*. Iwanami Publishing.

Daisuke Kawahara and Sadao Kurohashi. 1999. Corpus-based dependency analysis of Japanese sentences using verb bunsetsu transitivity. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium*, pages 387–391.

Daisuke Kawahara and Sadao Kurohashi. 2002. Fertilization of case frame dictionary for robust Japanese case analysis. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 425–431.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 2008–2013.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the Conference on Natural Language Learning*, pages 29–35.

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

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28.

Kiyoaki Shirai, 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 3rd Conference on Empirical Methods in Natural Language Processing*, pages 80–87.

Kentaro Torisawa. 2001. An unsupervised method for canonicalization of Japanese postpositions. In *Proceedings of the 6th Natural Language Processing Pacific Rim Simposium*, pages 211–218.

Kiyotaka Uchimoto, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara. 2000. Dependency model using posterior context. In *Proceedings of the 6th International Workshop on Parsing Technology*, pages 321–322.