

# Capturing Consistency between Intra-clause and Inter-clause Relations in Knowledge-rich Dependency and Case Structure Analysis

**Daisuke Kawahara**

National Institute of Information and  
Communications Technology,  
3-5 Hikaridai Seika-cho, Soraku-gun,  
Kyoto, 619-0289, Japan  
dk@nict.go.jp

**Sadao Kurohashi**

Graduate School of Informatics,  
Kyoto University,  
Yoshida-Honmachi, Sakyo-ku,  
Kyoto, 606-8501, Japan  
kuro@i.kyoto-u.ac.jp

## Abstract

We present a method for dependency and case structure analysis that captures the consistency between intra-clause relations (i.e., case structures or predicate-argument structures) and inter-clause relations. We assess intra-clause relations on the basis of case frames and inter-clause relations on the basis of transition knowledge between case frames. Both knowledge bases are automatically acquired from a massive amount of parses of a Web corpus. The significance of this study is that the proposed method selects the best dependency and case structure that are consistent within each clause and between clauses. We confirm that this method contributes to the improvement of dependency parsing of Japanese.

## 1 Introduction

The approaches of dependency parsing basically assess the likelihood of a dependency relation between two words or phrases and subsequently collect all the assessments for these pairs as the dependency parse of the sentence. To improve dependency parsing, it is important to consider as broad a context as possible, rather than a word/phrase pair.

In the recent evaluation workshops (shared tasks) of multilingual dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007), transition-based and graph-based methods achieved good performance by incorporating rich context. Transition-based dependency parsers consider the words following the word under consideration as features of machine learning (Kudo and Matsumoto, 2002; Nivre and Scholz, 2004; Sassano, 2004). Graph-based dependency parsers consider sibling and grandparent nodes,

i.e., second-order and higher-order features (McDonald and Pereira, 2006; Carreras, 2007; Nakagawa, 2007).

It is desirable to consider a wider-range phrase, clause, or a whole sentence, but it is difficult to judge whether the structure of such a wide-range expression is linguistically correct. One of the reasons for this is the scarcity of the knowledge required to make such a judgment. When we use the Penn Treebank (Marcus et al., 1993), which is one of the largest corpora among the available analyzed corpora, as training data, even bi-lexical dependencies cannot be learned sufficiently (Bikel, 2004). To circumvent such scarcity, for instance, Koo et al. (2008) proposed the use of word classes induced by clustering words in a large raw corpus. They succeeded in improving the accuracy of a higher-order dependency parser.

On the other hand, some researchers have proposed other approaches where linguistic units such as predicate-argument structures (also known as case structures and logical forms) are considered instead of arbitrary nodes such as sibling nodes. To solve the problem of knowledge scarcity, they learned knowledge of such predicate-argument structures from a very large number of automatically analyzed corpora (Abekawa and Okumura, 2006; Kawahara and Kurohashi, 2006b). While Abekawa and Okumura (2006) used only co-occurrence statistics of verbal arguments, Kawahara and Kurohashi (2006b) assessed predicate-argument structures by checking case frames, which are semantic frames that are automatically compiled for each predicate sense from a large raw corpus. These methods outperformed the accuracy of supervised dependency parsers.

In such linguistically-motivated approaches, well-formedness within a clause was considered, but coherence between clauses was not considered. Even if intra-clause relations (i.e., a predicate-argument structure within a clause) are

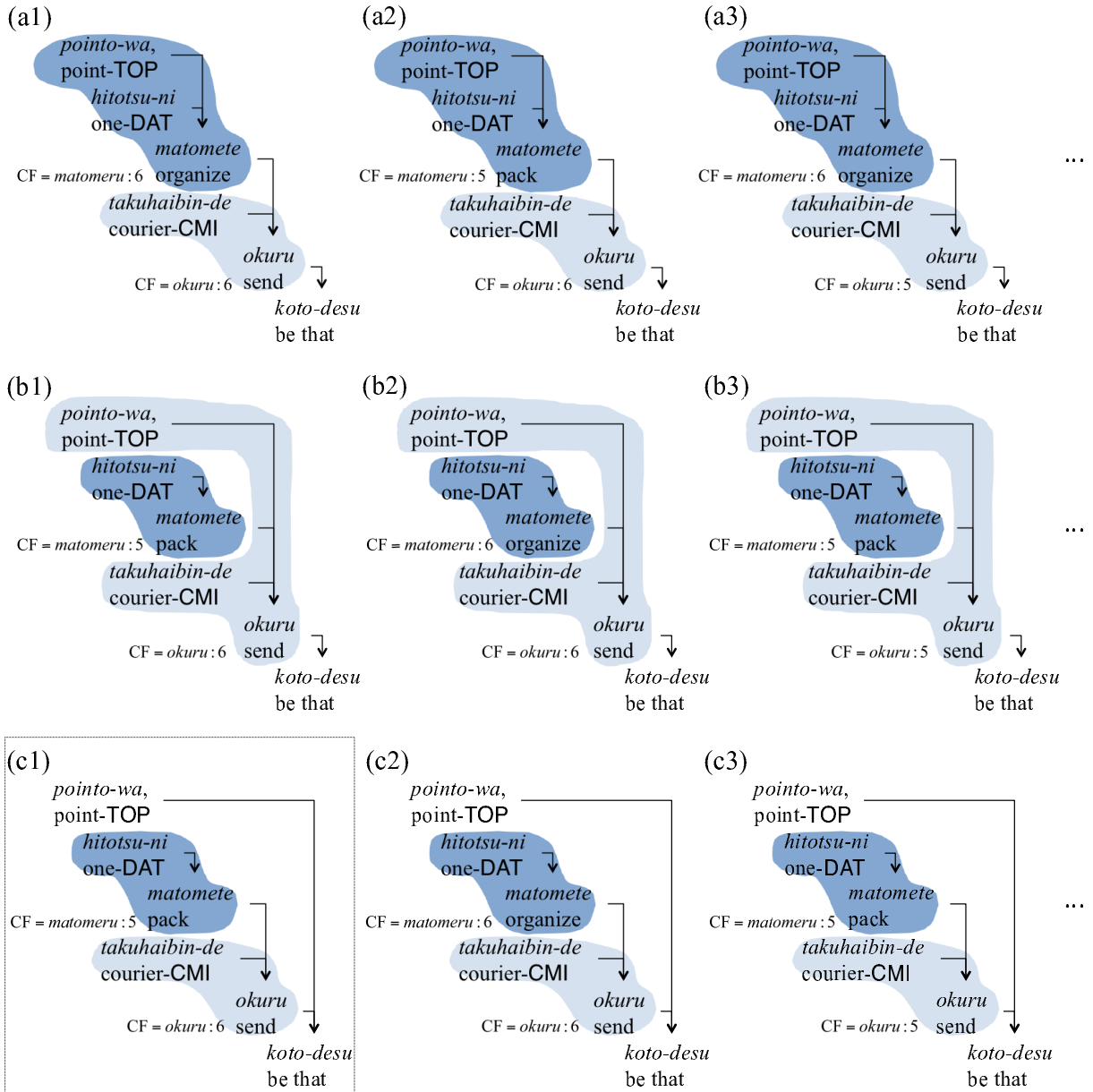


Figure 1: Possible dependency and case structures of sentence (1).

optimized, they might not be optimum when looking at clause pairs or sequences. To improve the accuracy of dependency parsing, we propose a method for dependency and case structure analysis that considers the consistency between intra-clause and inter-clause relations. This method analyzes intra-clause relations on the basis of *case frames* and inter-clause relations on the basis of *transition knowledge* between case frames. These two knowledge sources are automatically acquired from a massive amount of parses of a Web corpus.

The contributions of this paper are two-fold. First, we acquire transition knowledge not between verbs or verb phrases but between case

frames, which are semantically disambiguated representations. Second, we incorporate the transition knowledge into dependency and case structure analysis to capture the consistency between intra-clause and inter-clause relations.

The remainder of this paper is organized as follows. Section 2 illustrates our idea. Section 3 describes a method for acquiring the transition knowledge. Section 4 explains the proposed method of incorporating the acquired transition knowledge into a probabilistic model of dependency and case structure analysis. Section 5 reports experimental results. Section 6 gives the conclusions.

## 2 Idea of Capturing Consistency between Intra-clause and Inter-clause Relations

We propose a method for generative dependency parsing that captures the consistency between intra-clause and inter-clause relations.

Figure 1 shows the ambiguities of dependency and case structure of *pointo-wa* (point-TOP) in the following sentence:

- (1) *pointo-wa, hitotsu-ni matomete*  
point-TOP one-DAT pack

*takuhaibin-de okuru koto-desu*  
courier-CMI send be that

(The point is that (we) pack (one’s baggage) and send (it) using courier service.)

The correct structure is (c1), which is surrounded by the dotted rectangle. Structures (c2), (c3) and so on have the same dependency structure as (c1), but have incorrect case structures, in which incorrect case frames are selected. Note that *matomeru:5*, *okuru:6* and so on in the figure represent the IDs of the case frames.

The parser of Kawahara and Kurohashi (2006b) (and also conventional Japanese parsers) erroneously analyzes the head of *pointo-wa* (point-TOP)<sup>1</sup> as *matomete* (organize), whereas the correct head is *koto-desu* (be that), as shown in structure (a1) in Figure 1.

This error is caused by the incorrect selection of the case frame *matomeru:6* (organize), which is shown in Table 1. This case frame locally matches the input predicate-argument structure “*pointo-wa hitotsu-ni matomeru*” (organize points). Therefore, this method considers only intra-clause relations, and falls into local optimum.

If we consider the wide range of two clauses, this error can be corrected. In structure (a1) in Figure 1, the generative probability of case frame transition,  $P(\text{matomeru:6}|\text{okuru:6})$ , is considered. This probability value is very low, because there are few relations between the case frame *matomeru:6* (organize) and the case frame *okuru:6* (send baggage) in corpora.

Consequently, structure (c1) is chosen as the correct one, where both intra-clause and inter-clause relations can be interpreted by the *case*

<sup>1</sup>In this paper, we use the following abbreviations: NOM (nominative), ACC (accusative), ABL (ablative), CMI (comitative) and TOP (topic marker).

Table 1: Case frame examples for *matomeru* and *okuru*. “CS” represents case slot. Argument words are written only in English. “<num>” represents the class of numerals.

case frame ID	CS	example words
⋮	⋮	⋮
<i>matomeru:5</i> (pack)	<i>ga</i> <i>wo</i> <i>ni</i>	I, person, ... baggage, luggage, variables, ... <num>, pieces, compact, ...
<i>matomeru:6</i> (organize)	<i>ga</i> <i>wo</i> <i>ni</i>	doctor, ... point, singularity, ... <num>, pieces, below, ...
⋮	⋮	⋮
<i>okuru:1</i> (send)	<i>ga</i> <i>wo</i> <i>ni</i> <i>de</i>	person, I, ... mail, message, information, ... friend, address, direction, ... mail, post, postage, ...
⋮	⋮	⋮
<i>okuru:6</i> (send)	<i>ga</i> <i>wo</i> <i>ni</i> <i>de</i>	woman, ... baggage, supply, goods, ... person, Japan, parental house, ... mail, post, courier, ...
⋮	⋮	⋮

*frames* and the *transition knowledge* between case frames.

## 3 Acquiring Transition Knowledge between Case Frames

We automatically acquire large-scale transition knowledge of inter-clause relations from a raw corpus. The following two points are different from previous studies on the acquisition of inter-clause knowledge such as entailment/synonym knowledge (Lin and Pantel, 2001; Torisawa, 2006; Pekar, 2006; Zanzotto et al., 2006), verb relation knowledge (Chklovski and Pantel, 2004), causal knowledge (Inui et al., 2005) and event relation knowledge (Abe et al., 2008):

- the unit of knowledge is disambiguated and generalized

The unit in previous studies was a verb or a verb phrase, in which verb sense ambiguities still remain. Our unit is case frames that are semantically disambiguated.

- the variation of relations is not limited

Although previous studies focused on limited kinds of semantic relations, we comprehensively collect generic relations between clauses.

In this section, we first describe our unit of transition knowledge, case frames, briefly. We then detail the acquisition method of the transition knowledge, and report experimental results. Finally, we refer to related work to the acquisition of such knowledge.

### 3.1 The Unit of Transition Knowledge: Case Frames

In this paper, we regard case frames as the unit of transition knowledge. Case frames are constructed from unambiguous structures and are semantically clustered according to their meanings and usages. Therefore, case frames can be a less ambiguous and more generalized unit than a verb and a verb phrase. Due to these characteristics, case frames are a suitable unit for acquiring transition knowledge and weaken the influence of data sparseness.

#### 3.1.1 Automatic Construction of Case Frames

We employ the method of Kawahara and Kurohashi (2006a) to automatically construct case frames. In this section, we outline the method for constructing the case frames.

In this method, a large raw corpus is automatically parsed, and the case frames are constructed from argument-head examples in the resulting parses. The problems in automatic case frame construction are syntactic and semantic ambiguities. In other words, the parsing results inevitably contain errors, and verb senses are intrinsically ambiguous. To cope with these problems, case frames are gradually constructed from reliable argument-head examples.

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

Thereafter, the basic case frames are clustered in order to merge similar case frames, including similar case frames that are made from scrambled sentences. For example, since “*nimotsu-*

*wo tsumu*” (load baggage) and “*busshi-wo tsumu*” (load supply) are similar, they are clustered together. The similarity is measured by using a distributional thesaurus based on the study described in Lin (1998).

### 3.2 Acquisition of Transition Knowledge from Large Corpus

To acquire the transition knowledge, we collect the clause pairs in a large raw corpus that have a dependency relation and represent them as pairs of case frames. For example, from the following sentence, a case frame pair, (*matomeru:5, okuru:6*), is extracted.

- (2) *nimotsu-wo matomete, takuhaibin-de*  
 baggage-ACC pack courier-CMI  
*okutta*  
 sent

(packed one’s baggage and sent (it) using courier service)

These case frames are determined by applying a conventional case structure analyzer (Kawahara and Kurohashi, 2006b), which selects the case frames most similar to the input expressions “*nimotu-wo matomeru*” (pack baggage) and “*takuhaibin-de okuru*” (send with courier service) from among the case frames of *matomeru* (organize/settle/pack/...) and *okuru* (send/remit/see off/...); some of the case frames of *matomeru* and *okuru* are listed in Table 1.

We adopt the following steps to acquire the transition knowledge between case frames:

1. Apply dependency and case structure analysis to assign case frame IDs to each clause in a large raw corpus.
2. Collect clause pairs that have a dependency relation, and represent them as pairs of case frame IDs.
3. Count the frequency of each pair of case frame IDs; these statistics are used in the analysis described in Section 4.

At step 2, we collect both syntactically ambiguous and unambiguous relations in order to alleviate data sparseness. The influence of a small number of dependency parsing errors would be hidden by a large number of correct (unambiguous) relations.

Table 2: Examples of automatically acquired transition knowledge between case frames.

pairs of case frame IDs	meaning	freq.
( <i>okuru</i> :1, <i>okuru</i> :6)	(send mails, send baggage)	186
( <i>aru</i> :1, <i>okuru</i> :6)	(have, send baggage)	150
( <i>suru</i> :1, <i>okuru</i> :6)	(do, send baggage)	134
( <i>issyoda</i> :10, <i>okuru</i> :6)	(get together, send baggage)	118
( <i>kaku</i> :1, <i>okuru</i> :6)	(write, send baggage)	115
⋮	⋮	⋮
( <i>matomeru</i> :5, <i>okuru</i> :6)	(pack, send baggage)	12
( <i>dasu</i> :3, <i>okuru</i> :6)	(indicate, send baggage)	12
⋮	⋮	⋮

### 3.3 Experiments of Acquiring Transition Knowledge between Case Frames

To obtain the case frames and the transition knowledge between case frames, we first built a Japanese Web corpus by using the method of Kawahara and Kurohashi (2006a). We first crawled 100 million Japanese Web pages, and then, we extracted and unduplicated Japanese sentences from the Web pages. Consequently, we developed a Web corpus consisting of 1.6 billion Japanese sentences.

Using the procedure of case frame construction presented in Section 3.1.1, we constructed case frames from the whole Web corpus. They consisted of 43,000 predicates, and the average number of case frames for a predicate was 22.2.

Then, we acquired the transition knowledge between case frames using 500 million sentences of the Web corpus. The resulting knowledge consisted of 108 million unique case frame pairs. Table 2 lists some examples of the acquired transition knowledge. In the acquired transition knowledge, we can find various kinds of relation such as entailment, cause-effect and temporal relations.

Let us compare this result with the results of previous studies. For example, Chklovski and Pantel (2004) obtained 29,165 verb pairs for several semantic relations in VerbOcean. The transition knowledge acquired in this study is several thousand times larger than that in VerbOcean. It is very difficult to make a meaningful comparison, but it can be seen that we have succeeded in acquiring generic transition knowledge on a large scale.

### 3.4 Related Work

In order to realize practical natural language processing (NLP) systems such as intelligent dialog

systems, a lot of effort has been made to develop world knowledge or inference knowledge. For example, in the CYC (Lenat, 1995) and Open Mind (Stork, 1999) projects, such knowledge has been obtained manually, but it is difficult to manually develop broad-coverage knowledge that is sufficient for practical use in NLP applications.

On the other hand, the automatic acquisition of such inference knowledge from corpora has attracted much attention in recent years. First, semantic knowledge between entities has been automatically obtained (Girju and Moldovan, 2002; Ravichandran and Hovy, 2002; Pantel and Pennacchiotti, 2006). For example, Pantel and Pennacchiotti (2006) proposed the Espresso algorithm, which iteratively acquires entity pairs and extraction patterns using reciprocal relationship between entities and patterns.

As for the acquisition of the knowledge between events or clauses, which is most relevant to this study, many approaches have been adopted to acquire entailment knowledge. Lin and Pantel (2001) and Szpektor and Dagan (2008) learned entailment rules based on distributional similarity between instances that have a relation to a rule. Torisawa (2006) extracted entailment knowledge using coordinated verb pairs and noun-verb co-occurrences. Pekar (2006) also collected entailment knowledge with discourse structure constraints. Zanzotto et al. (2006) obtained entailment knowledge using nominalized verbs.

There have been some studies on relations other than entailment relations. Chklovski and Pantel (2004) obtained verb pairs that have one of five semantic relations by using a search engine. Inui et al. (2005) classified the occurrences of the Japanese connective marker *tame*. Abe et al.

(2008) learned event relation knowledge for two semantic relations. They first gave seed pairs of verbs or verb phrases and extracted the patterns that matched these seed pairs. Subsequently, by using the Espresso algorithm (Pantel and Pennacchiotti, 2006), this process was iterated to augment both instances and patterns. The acquisition unit in these studies was a verb or a verb phrase.

In contrast to these studies, we obtained generic transition knowledge between case frames without limiting target semantic relations.

#### 4 Incorporating Transition Knowledge into Dependency and Case Structure Analysis

We employ the probabilistic generative model of dependency and case structure analysis (Kawahara and Kurohashi, 2006b) as a base model. We incorporate the obtained transition knowledge into this base parser.

Our model assigns a probability to each possible dependency structure,  $T$ , and case structure,  $L$ , of the input sentence,  $S$ , and outputs the dependency and case structure that have the highest probability. In other words, the model selects the dependency structure  $T_{best}$  and the case structure  $L_{best}$  that maximize the probability  $P(T, L|S)$  or its equivalent,  $P(T, L, S)$ , as follows:

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

The last equation follows from the fact that  $P(S)$  is constant.

In the model, a clause (or predicate-argument structure) is considered as a generation unit and the input sentence is generated from the end of the sentence. The probability  $P(T, L, S)$  is defined as the product of the probabilities of generating clauses  $C_i$  as follows:

$$P(T, L, S) = \prod_{C_i \in S} P(C_i|C_h), \quad (2)$$

where  $C_h$  is the modifying clause of  $C_i$ . Since the Japanese language is head final, the main clause at the end of a sentence does not have a modifying head; we account for this by assuming  $C_h = \text{EOS}$  (End Of Sentence).

The probability  $P(C_i|C_h)$  is defined in a manner similar to that in Kawahara and Kurohashi

(2006b). However, the difference between the probability in the above-mentioned study and that in our study is the generative probability of the case frames, i.e., the probability of generating a case frame  $CF_i$  from its modifying case frame  $CF_h$ . The base model approximated this probability as the product of the probability of generating a predicate  $v_i$  from its modifying predicate  $v_h$  and the probability of generating a case frame  $CF_i$  from the predicate  $v_i$  as follows:

$$\begin{aligned} P(CF_i|CF_h) &\approx \\ &P(v_i|v_h) \times P(CF_i|v_i). \end{aligned} \quad (3)$$

Our proposed model directly estimates the probability  $P(CF_i|CF_h)$  and considers the transition likelihood between case frames. This probability is calculated from the transition knowledge between case frames using maximum likelihood.

In practice, to avoid the data sparseness problem, we interpolate the probability  $P(CF_i|CF_h)$  with the probability of generating predicates,  $P(v_i|v_h)$ , as follows:

$$\begin{aligned} P'(CF_i|CF_h) &\approx \\ &\lambda P(CF_i|CF_h) + (1 - \lambda)P(v_i|v_h), \end{aligned} \quad (4)$$

where  $\lambda$  is determined using the frequencies of the case frame pairs,  $(CF_i, CF_h)$ , in the same manner as in Collins (1999).

## 5 Experiments

We evaluated the dependency structures that were output by our new dependency parser. The case frames used in these experiments are the same as those described in Section 3.3, which were automatically constructed from 1.6 billion Japanese sentences obtained from the Web.

In this study, the parameters related to unlexical types were calculated from the Kyoto University Text Corpus, which is a small tagged corpus of newspaper articles, and lexical parameters were obtained from a large Web corpus. To evaluate the effectiveness of our model, our experiments were conducted using sentences obtained from the Web. As a test corpus, we used 759 Web sentences<sup>2</sup>, which were manually annotated using the same criteria as those in the case of the Kyoto University Text Corpus. We also used the Kyoto University Text Corpus as a development corpus to optimize some smoothing parameters. The system

<sup>2</sup>The test set was not used to construct case frames and estimate probabilities.

Table 3: The dependency accuracies in our experiments.

	syn	syn+case	syn+case+cons
all	4,555/5,122 (88.9%)	4,581/5,122 (89.4%)	4,599/5,122 (89.8%)
NP→VP	2,115/2,383 (88.8%)	2,142/2,383 (89.9%)	2,151/2,383 (90.3%)
NP→NP	1,068/1,168 (91.4%)	1,068/1,168 (91.4%)	1,068/1,168 (91.4%)
VP→VP	779/928 (83.9%)	777/928 (83.7%)	783/928 (84.4%)
VP→NP	579/623 (92.9%)	579/623 (92.9%)	582/623 (93.4%)

input was automatically tagged using the JUMAN morphological analyzer<sup>3</sup>.

We used two baseline systems for the purposes of comparison: a rule-based dependency parser (Kurohashi and Nagao, 1994) and the probabilistic generative model of dependency and case structure analysis (Kawahara and Kurohashi, 2006b)<sup>4</sup>. We use the above-mentioned case frames also in the latter baseline parser, which also requires automatically constructed case frames.

### 5.1 Evaluation of Dependency Structures

We evaluated the obtained dependency structures in terms of phrase-based dependency accuracy — the proportion of correct dependencies out of all dependencies<sup>5</sup>.

Table 3 lists the dependency accuracies. In this table, “syn” represents the rule-based dependency parser, “syn+case” represents the probabilistic parser of syntactic and case structure (Kawahara and Kurohashi, 2006b)<sup>6</sup>, and “syn+case+cons” represents our proposed model. In the table, the dependency accuracies are classified into four categories on the basis of the phrase classes (VP: verb phrase<sup>7</sup> and NP: noun phrase) of a dependent and its head. The parser “syn+case+cons” significantly outperformed the two baselines for “all” (McNemar’s test;  $p < 0.05$ ). In particular, the accuracy of the intra-clause (predicate-argument) relations (“NP→VP”) was improved by 1.5% from “syn” and by 0.4% from “syn+case.” These im-

<sup>3</sup><http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman-e.html>

<sup>4</sup><http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp-e.html>

<sup>5</sup>Since Japanese is head-final, the second to last phrase unambiguously depends on the last phrase. However, we include such dependencies into our evaluation as in most of previous studies.

<sup>6</sup>The accuracy described in Kawahara and Kurohashi (2006b) is different from that of this paper due to the different evaluation measure excluding the unambiguous dependencies of the second last phrases.

<sup>7</sup>VP includes not only verbs but also adjectives and nouns with copula.

provements are due to the incorporation of the transition knowledge into syntactic/case structure analysis.

In order to compare our results with a state-of-the-art discriminative dependency parser, we input the test corpus into an SVM-based Japanese dependency parser, CaboCha<sup>8</sup>(Kudo and Matsumoto, 2002), which was trained using the Kyoto University Text Corpus. Its dependency accuracy was 88.6% (4,540/5,122), which is close to that of “syn.” This low accuracy is attributed to the lack of knowledge of both intra-clause and inter-clause relations. Another cause of the low accuracy is the out-of-domain training corpus. In other words, the parser was trained on a newspaper corpus, while the test corpus was obtained from the Web because a tagged Web corpus that is large enough to train a supervised parser is not available.

### 5.2 Discussions

Figure 2 shows some improved analyses; here, the dotted lines represent the results of the analysis performed using the baseline “syn + case,” and the solid lines represent the analysis performed using the proposed method, “syn+case+cons.” These sentences are incorrectly analyzed by the baseline but correctly analyzed by the proposed method. For example, in sentence (a), the head of *gunegunemichi-wo* (winding road-ACC) was correctly analyzed as *yurareru* (be jolted). This is because the case frame of “*basu-ni yurareru*” (be jolted by bus) is likely to generate *tatsu* (stand) that does not take the *wo* (ACC) slot. In this manner, by considering the transition knowledge between case frames, the selection of case frames became accurate, and thus, the accuracy of the dependencies within clauses (predicate-argument structures) was improved.

In the case of the dependencies between predicates (VP→VP), however, only small improve-

<sup>8</sup><http://chasen.org/~taku/software/cabocha/>

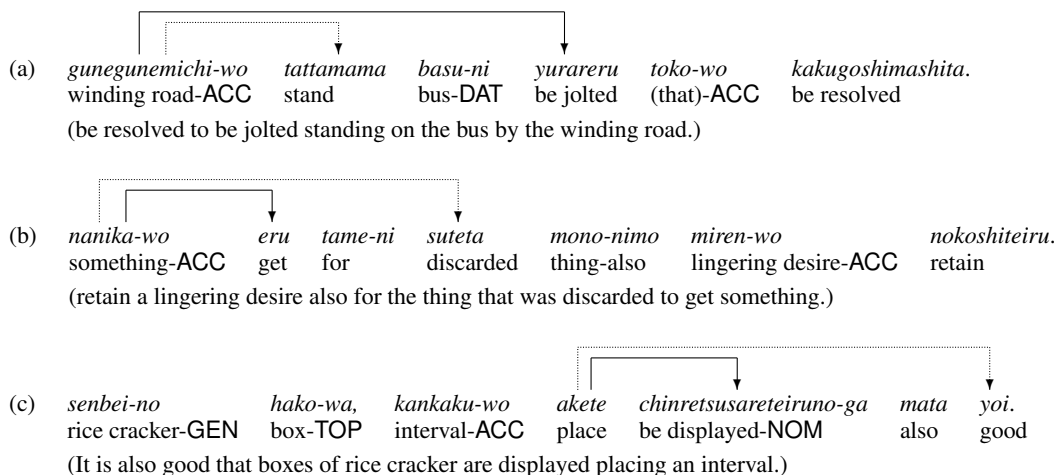


Figure 2: Improved examples.

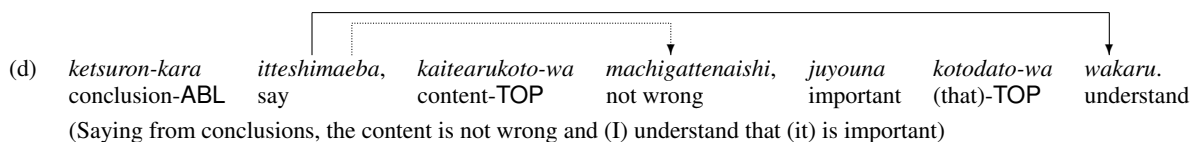


Figure 3: An erroneous example.

ments were achieved by using the transition knowledge between case frames. This is mainly because the heads of the predicates are intrinsically ambiguous in many cases.

For example, in sentence (d) in Figure 3, the correct head of *itteshimaeba* (say) is *wakaru* (understand) as designated by the solid line, but our model incorrectly judged the head to be *machigattenaiishi*, (not wrong) as designated by the dotted line. However, in this case, both the phrases that are being modified are semantically related to the modifier. To solve this problem, it is necessary to re-consider the evaluation metrics of dependency parsing.

## 6 Conclusion

In this paper, we have described a method for acquiring the transition knowledge of inter-clause relations and a method for incorporating this knowledge into dependency and case structure analysis. The significance of this study is that the proposed parsing method selects the best dependency and case structures that are consistent within each clause and between clauses. We confirmed that this method contributed to the improvement of the dependency parsing of Japanese.

The case frames that are acquired from 1.6 billion Japanese sentences have been made freely

available to the public<sup>9</sup>. In addition, we are preparing to make the acquired transition knowledge accessible on the Web.

In future, we will investigate the iteration of knowledge acquisition and parsing based on the acquired knowledge. Since our parser is a generative model, we are expecting a performance gain by the iteration. Furthermore, we would like to explore the use of the transition knowledge between case frames to improve NLP applications such as recognizing textual entailment (RTE) and sentence generation.

## References

- Shuya Abe, Kentaro Inui, and Yuji Matsumoto. 2008. Acquiring event relation knowledge by learning cooccurrence patterns and fertilizing cooccurrence samples with verbal nouns. In *Proceedings of IJCNLP2008*, pages 497–504.
- Takeshi Abekawa and Manabu Okumura. 2006. Japanese dependency parsing using co-occurrence information and a combination of case elements. In *Proceedings of COLING-ACL2006*, pages 833–840.
- Daniel M. Bikel. 2004. Intricacies of Collins’ parsing model. *Computational Linguistics*, 30(4):479–511.

<sup>9</sup><http://nlp.kuee.kyoto-u.ac.jp/nl-resource/casframe-e.html>



- Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*, pages 149–164.
- Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of EMNLP-CoNLL2007 Shared Task*, pages 957–961.
- Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP2004*, pages 33–40.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Roxana Girju and Dan Moldovan. 2002. Mining answers for causation questions. In *Proceedings of AAAI Spring Symposium*.
- Takashi Inui, Kentaro Inui, and Yuji Matsumoto. 2005. Acquiring causal knowledge from text using the connective marker tame. *ACM Transactions on Asian Language Information Processing (ACM-TALIP)*, 4(4):435–474.
- Daisuke Kawahara and Sadao Kurohashi. 2006a. Case frame compilation from the web using high-performance computing. In *Proceedings of LREC2006*.
- Daisuke Kawahara and Sadao Kurohashi. 2006b. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of HLT-NAACL2006*, pages 176–183.
- Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08:HLT*, pages 595–603.
- Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL2002*, 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.
- Douglas B. Lenat. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):32–38.
- Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL98*, pages 768–774.
- Mitchell Marcus, Beatrice Santorini, and Mary Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EAACL2006*, pages 81–88.
- Tetsuji Nakagawa. 2007. Multilingual dependency parsing using global features. In *Proceedings of EMNLP-CoNLL2007 Shared Task*, pages 952–956.
- Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING2004*, pages 64–70.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL2007*, pages 915–932.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of COLING-ACL2006*, pages 113–120.
- Viktor Pekar. 2006. Acquisition of verb entailment from text. In *Proceedings of HLT-NAACL2006*, pages 49–56.
- Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL2002*, pages 41–47.
- Manabu Sassano. 2004. Linear-time dependency analysis for Japanese. In *Proceedings of COLING2004*, pages 8–14.
- David G. Stork. 1999. Character and document research in the open mind initiative. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1–12.
- Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of COLING2008*, pages 849–856.
- Kentaro Torisawa. 2006. Acquiring inference rules with temporal constraints by using Japanese coordinated sentences and noun-verb co-occurrences. In *Proceedings of HLT-NAACL2006*, pages 57–64.
- Fabio Massimo Zanzotto, Marco Pennacchiotti, and Maria Teresa Pazienza. 2006. Discovering asymmetric entailment relations between verbs using selectional preferences. In *Proceedings of COLING-ACL2006*, pages 849–856.