Integration of Dependency Analysis with Semantic Analysis Referring to the Context

Yuki Ikegaya* **Yasuhiro Noguchi*** **Satoru Kogure†**
Tatsuhiro Konishi† **Makoto Kondo†** **Asoh Hideki‡**
Akira Takagi§ **Yukihiro Itoh†**

**Graduate School of Science and Engineering, Shizuoka University*
{cs7005@cs.,noguchi@}inf.shizuoka.ac.jp

†Faculty of Informatics, Shizuoka University
{kogure@cs,mkondo@ia,konishi@cs,itoh@cs}.inf.shizuoka.ac.jp

‡Information Technology Research Institute, AIST
h.asoh@aist.go.jp

§CSK Corporation/ Information Technology Research Institute, AIST
akira-takagi@aist.go.jp

Abstract

This paper describes how to perform syntactic parsing and semantic analysis using the contextual information as a language understanding component in a dialog system. Although syntactic parsing and semantic analysis are often conducted independently of each other, correct parsing of a sentence often requires the semantic information on the input and/or the contextual information prior to the input. We therefore merge syntactic parsing with semantic analysis, which enables syntactic parsing to take advantage of the semantic content of an input and its contextual information. To use contextual information, the semantic representation of an input should have a comparable form to the semantic content of the preceding context. Accordingly, we employ a framework for semantic representations that achieves such comparison. We take dialogs of hotel search and reservation for example, and demonstrate the effectiveness of the proposed method. The experimental results confirm that the proposed system achieves high accuracy in parsing and generation of semantic representations.

1. Introduction

When we have a talk, we presuppose the situation and particular context.[1] The situation/context enables us to understand utterances without talking in great detail. The situation/context also makes it possible to convey a wide variety of contents with a certain linguistic expression. This means that a language understanding component in a dialog system should not interpret a given sentence alone; rather, it should interpret a sentence using the context information.

In our previous work, we have developed a Japanese dialog system for hotel search and reservation (Noguchi et al. 2002). The system accepts free input from the keyboard. We are planning to build an audio input module into the system. Audio input raises the importance of the context information because the effective use of the context information may circumvent possible recognition errors. In this paper, we focus on how a language understanding component can perform syntactic parsing and semantic analysis using the contextual information.

It may be possible to integrate syntactic parsing with semantic analysis by taking advantage of LFG (Kaplan et al. 1982) and HPSG （Pollard et al. 1994). However, they do not consider the use of the context information. In this paper, we propose the method to integrate dependency analysis with semantic analysis referring to the context.

Recently, many researches have adopted a statistical approach toward parsing, and it has been shown that the statistical approach is effective in obtaining correct dependency structure. There are

[1] The term “context” primarily means association of utterances. In particular, it refers to logical relationship among words and/or sentences. In this paper, the term is used to indicate the interpretive results of input/output sentences and conceptual knowledge about the words included there.

some free tools available (e.g. Charniak 2000; Kurohashi 1998; Kudo et al. 2002). In the study of semantic analysis, word sense disambiguation (Charniak 2000) and analysis of case (Harada et al. 2000) have been pursued. The syntactic parsing and semantic analysis, however, are often studied independently of each other though correct parsing of a sentence often requires the semantic information on the input and/or the contextual information prior to the input. Accordingly, we propose a sequential parser that determines syntactic structure based on semantic analysis referring to the context and on relevant statistical information.

2. Fundamental Considerations

2.1.Syntactic and Semantic analysis

Japanese has the property of free word order and it raises the number of possible candidates in parsing a given sentence. If parsing and semantic analysis are performed independently, it is difficult to efficiently reduce the number of the candidates using the semantic/contextual information. If semantic analysis applies to every candidate, the calculation cost may be too high.

To reduce the number of candidates using the contextual information, the system process an input word by word and, in each cycle of parsing, it gives a score to each candidate based on grammatical rules and the conformity to the context. In each cycle of parsing, the system generates possible semantic representations by referring to the context. Each representation receives a score and low-scored candidates are filtered out. To use contextual information in the process, the contextual information has to be stored in such a way that it can be formally compared to the semantic content of an input. For example, the system should realize whether an input contains something that has already been denoted in the context. If it does, the system should realize which part in the input has appeared in the context and how that part has been mentioned.

The next subsection describes the framework for semantic representations which is capable of extracting the semantic content from the context.

2.2.Semantic Representations

Conventional semantic representations do not achieve the goal mentioned above because they inherit syntactic dependency and have as much variety as dependency structures. Take the following sentences for example.

(1-a) *watasi-wa Tokyo-kara Hamamatsu-no daigaku-ni iku*. (I will go from Tokyo to a university in Hamamatsu.) -> Go (I, Tokyo, a university in Hamamatsu)

(1-b) *watasi-wa Hamamatsu-no daigaku-ni iku. shuppatsuchi-wa Tokyo da*. (I will go to a university in Hamamatsu. The departure place is Tokyo.) -> Go (I, NIL, a university in Hamamatsu) and Eq (the departure place, Tokyo)

(1-a) and (1-b) convey roughly the same meaning. Predicate logic gives the two examples the logical forms on the right of the arrowheads, respectively. Transformation from the natural language expressions to the logical forms retains the same dependency. The dependency between *iku* and the three words, *watashi, Tokyo* and *Hamamatsu-no daigaku,* is identical to the one between Go and the arguments. The same is equally true of (1-b). In order to judge to the equivalence of (1-a) with (1-b), we need the following specific interpretive rules.

The second argument of Go is the value of a departure place. If the value of the first argument of Eq is a departure place, the value of the second argument of Go and the one in the second argument of Eq are equivalent.

It, however, is not practical give such rules to every type of dependency structures. Semantic representations, therefore, should enable formal evaluation meanings irrespective of the number of sentences to convey a semantic content and the evaluation should not be affected by the types of dependency structures involved in the input.

In short, semantic representations should allow the system to synthesize meanings of several sentences as if they were conveyed in one sentence and they should also enable comparison of semantic contents of sentences. Our semantic representations meet the requirements above.

Within our framework, any dependency structures in an input sentence are transformed into a single type of dependency structure which contains only one type of predicate, the identifying predicate da/de-aru (a Japanese counterpart to English be). The transformation standardizes any dependency into one between the identifying predicate and its arguments. Semantic representations are derived by transforming every dependency structure in this manner. The transformation is applicable not only to phrase with case particles but also to other kinds of phrases (Takagi et al. 1987).

When a noun is modified by a phrase with an adnominal particle, the entire phrase can be transformed into the form with the predicate *da/de-aru*: “an attribute of the entity is its value”. Adjectives and adverbs have the property to designate the value of some attribute involved in an entity and an event, respectively in Figure 1. Consequently, dependency structures containing them can be transformed into the form “an attribute is a value” in Figure 2. The sentence in (1-a), for example, is transformed as in Figure. 3.

We have defined concepts which would appear in dialogs for our system (i.e. hotel search/reservation). The definition is based on example dialogs, books, etc. We have examined what kinds of entities and events would appear in the relevant domain, and what types of attributes should be defined for the entities and the predicates denoting the events. The attribute nouns are put into a concept hierarchy based on their super-sub/whole-part relations. Comparison of meanings of phrases/clauses is performed by comparing attribute nouns based on the concept hierarchy.
Figure 4 shows the semantic representations of (1-a) and (1-b). The comparison between the attribute-value pairs enables evaluation of the synonymity without any specific interpretive rules.
Although the value of the event attribute *ido-kiten* (source of movement) in the semantic representation of *ido-koui* (Go) is NIL in (1-b), the system identifies it as Tokyo based on the super-sub relation between *ido-kiten* and *shuppatsuchi*. Accordingly, the value of *ido-kiten* in (1-b) is judged as equivalent to the one in (1-a).

By comparing every attribute-value pair in this manner, (1-a) is judged as equivalent to (1-b). The comparison of attribute-value pairs is not affected by whether the relevant attributes are mentioned in one sentence or in more than one sentence. Thus, the interpretation is not affected by the number of sentences that denotes a certain semantic content.

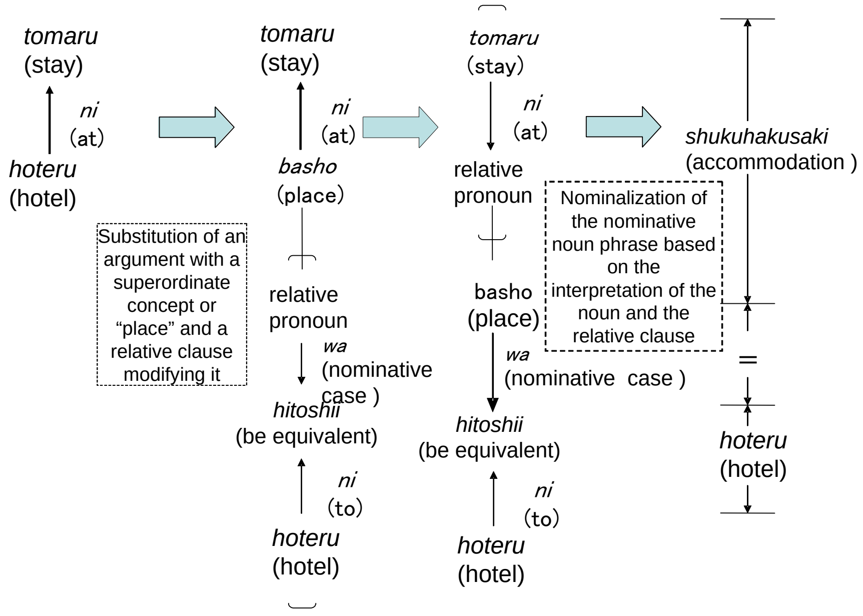


Figure1: Transformation of dependency structure

event/entity attribute 1	value 1
event/entity attribute 2	value 2
. . .	
event/entity attribute *m*	value *m*

Figure2: Semantic Representation of Predicate

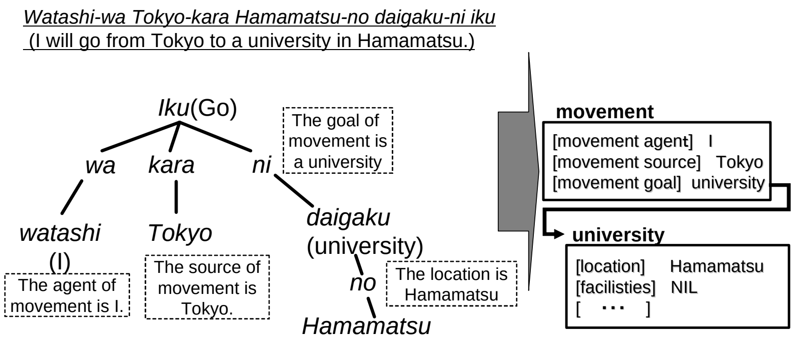

Watashi-wa Tokyo-kara Hamamatsu-no daigaku-ni iku
(I will go from Tokyo to a university in Hamamatsu.)
Iku(Go)
wa
kara
ni
watashi (I)
Tokyo
daigaku (university)
no
Hamamatsu
The agent of movement is I.
The source of movement is Tokyo.
The goal of movement is a university
The location is Hamamatsu
movement
[movement agent] I
[movement source] Tokyo
[movement goal] university
university
[location] Hamamatsu
[facilisties] NIL
[···]

Figure 3: Transformation to Semantic representation

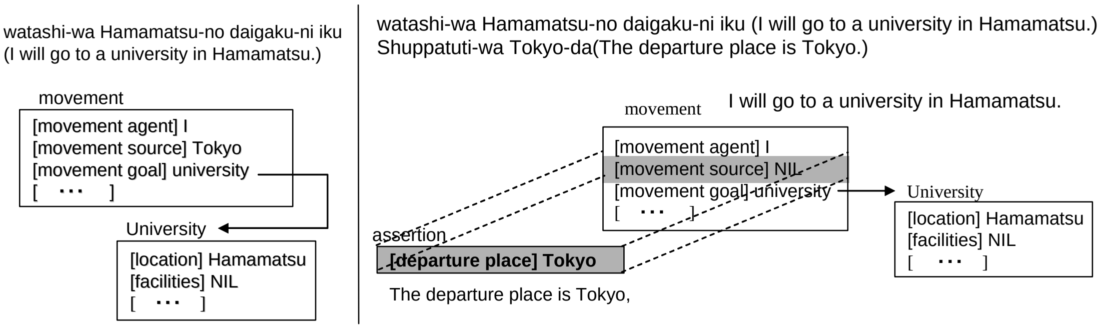

watashi-wa Hamamatsu-no daigaku-ni iku
(I will go to a university in Hamamatsu.)
movement
[movement agent] I
[movement source] Tokyo
[movement goal] university
[···]
University
[location] Hamamatsu
[facilities] NIL
[···]
watashi-wa Hamamatsu-no daigaku-ni iku (I will go to a university in Hamamatsu.)
Shuppatuti-wa Tokyo-da(The departure place is Tokyo.)
movement
I will go to a university in Hamamatsu.
[movement agent] I
[movement source] NIL
[movement goal] university
[···]
University
[location] Hamamatsu
[facilities] NIL
[···]
assertion
[departure place] Tokyo
The departure place is Tokyo,

Figure 4: Semantic Comparison

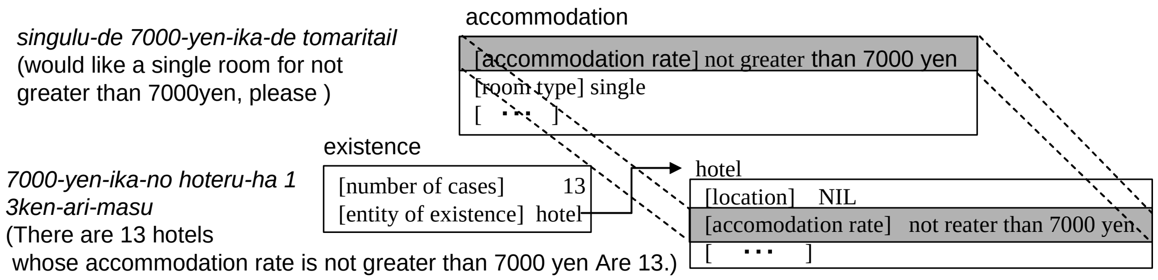
2.3.Reference to the Context

The comparison of attribute-value pairs achieves the reference to the context.

Suppose that we compare a pair "attribute A = a" with another pair "attribute B = b". When the attribute A and the value a are identical with B and b, respectively, the pairs are judged as denoting the same meaning, which is illustrated in Figure 5.

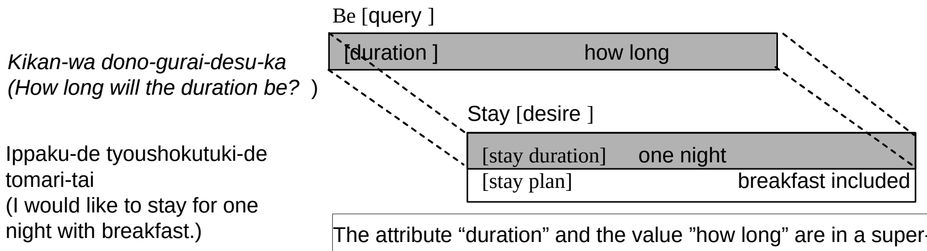
Figure 6 shown the case in which the attribute A, and the value are in super-sub relation with B and b, respectively. In this case, the input is regarded as relevant to the context. When one attribute is a primary attribute and there is another attribute which is a sub-attribute to the former in Figure 6, the input is judged as consistent to the context. [2]

[2] When the value of an attribute is dependent on the value of another attribute, we call the former a primary attribute and the latter a sub-attribute. If hotel rates are dependent on the room type and on whether breakfast is included, the attribute "hotel rate" is a primary attribute and the attributes "room type" and "accommodation plan" are sub-attributes. If an input contains either a primary attribute or the sub-attribute and the context involves the other, the input can be regarded as involving something that affects or is affected by the context.


singulu-de 7000-yen-ika-de tomaritaiI
(would like a single room for not greater than 7000yen, please)
accommodation
[accommodation rate] not greater than 7000 yen
[room type] single
[･･･]
existence
7000-yen-ika-no hoteru-ha 1 3ken-ari-masu
(There are 13 hotels whose accommodation rate is not greater than 7000 yen Are 13.)
[number of cases] 13
[entity of existence] hotel
hotel
[location] NIL
[accomodation rate] not reater than 7000 yen
[･･･]

-The attribute pairs are identical, “[accommodation rate] not greater than 7000 yen” of accomodation and hotel.
-“[accommodation rate] not greater than 7000 yen” is a primary attribute and “[room type] single” is a sub-attribute to the former.

Figure 5: Comparison of attribute-value pairs (1)


Be [query]
[duration] how long
Kikan-wa dono-gurai-desu-ka
(How long will the duration be?)
Stay [desire]
[stay duration] one night
[stay plan] breakfast included
Ippaku-de tyoushokutuki-de tomari-tai
(I would like to stay for one night with breakfast.)

The attribute “duration” and the value ”how long” are in a super-sub relation with the attribute “stay duration” and the value “one night”, respectively.

Figure 6: Comparison of attribute-value pairs (2)

In referring to the context, we should delimit the scope of the reference. This is because the calculation cost for the search in the context may be extremely high if the entire context is always the scope of reference.

In general, a phrase with a topic particle -wa or a nominative particle –wa indicates a shift of topic, and the generalization holds true for dialogs about hotel search and reservation (Noguchi et al. 2002).

Accordingly, reference to the context goes back to either of the followings.

To a word with a topic particle -wa

To a word with a nominative particle –wa

When the last word with –wa in the context (W1) and the word with –wa immediately preceding W1-are in a whole-part relation where W2 contains W1, the scope is extended to the sentence containing W2.

3. Integration of Syntactic and Semantic Analyses

3.1.Processing algorithm

Syntactic and semantic analysis proceeds from the sentence initial morpheme to the sentence final one. The system picks up a morpheme and retrieves the relevant concept from a frame dictionary which contains frames associated with concepts. Figure 7 is the entire processing flow.

Frames in the frame dictionary contain knowledge in the form in Figure 1, lexical entries, parts of speech, conjugations, and classes of concepts. Syntactic and semantic analysis processes all concepts by using stacks. A retrieved concept receives parsing and semantic analysis with the top concept in a stack. The parsing and semantic analysis are recursively applied to every element in the stack to generate partial dependency trees and associated semantic representations. Since crossing dependency is generally prohibited, if the retrieved concept does not have dependency with some concept in the stack, the process in the stack is finished there.

Syntactic parsing uses rules of dependency grammar based on parts of speech. When a retrieved concept and the top concept in a partial tree may have dependency from the syntactic view point, they are given semantic analysis.

Semantic analysis is based on semantic selectional restrictions and determines in which attribute in the semantic representation the dependent concept should be stored.

If there are several candidate attributes to store the dependent concept, every candidate tree/semantic representation is generated separately. In order to avoid the explosion of candidates, attribute-value pairs generated from a retrieved concept is compared with attribute value pairs in partial trees in the stack and the context. The comparison with partial trees is performed only if the retrieved concept has dependency with the tree. The reference to the context goes from the last sentence to the beginning of the dialogue.

A candidate semantic representation receives a point if there is an attribute-value pair relevant to the generated attribute-value pair in the context. The process proceeds to the end of the sentence word by word and generates semantic representations, retaining candidates given higher scores.

Relevant statistical information is extracted from the corpus dialogs about hotel search and reservation (200 sentences, 22 dialogs) by examining what kinds of dependencies are included in the corpus. First, we investigated which content word depends on which content word via which kind of dependency. The statistical information used in our system is derived from the following equation, where *hi*, *hj* and *ri* indicate the root form of a depending content word, the root form of a depended content word and the type of dependency between them, respectively.

$$P(hi \xrightarrow{ri} hj \mid hi, ri) = \frac{C(hi \xrightarrow{ri} hj, hi, hj, ri)}{\sum_x C(hi, x, ri)}$$

The equation gives us the probability that *hj* is the depended word when *hi* depends on another content word in the dependency *ri*. The uses this information every time it generates a dependency between content words. The relevant score is raised if the dependency has the probability greater than a designated threshold.

Table 1: Types of dependencies

a depending word has an adnominal form
a depending word has an adverbial form
a dependency via a conjunctive particle
a depending word is an adnominal word
dependencies via case particles (*wa, ga, kara, made, de, e, ni*)
a dependency via an adnominal particle *no*

Figure8 shows generation process of semantic representations for the sentence *keshiki-no yoi-hoteru-ga ii-na* (I prefer a scenic hotel).Although the representations are depicted in a simply form, the elements in the stacks are actually semantic representations in the form of attribute-value pairs. After the string keshiki-no yoi (be scenic) is processed, the candidates (a) and (b) are generated. *Kesiki-no* modifies *yoi* in (b) but it doesn't in (a). Then, *hoteru* (hotel) is analyzed. If grammatical rules judges that *hoteru* may modify the top element in the stacks (*yoi*) they may have dependency. Therefore, the process generates candidates in which *hoteru* modifies *yoi* and candidates where it doesn't. If a couple of concepts may modify a retrieved concept, the process above is repeated recursively.

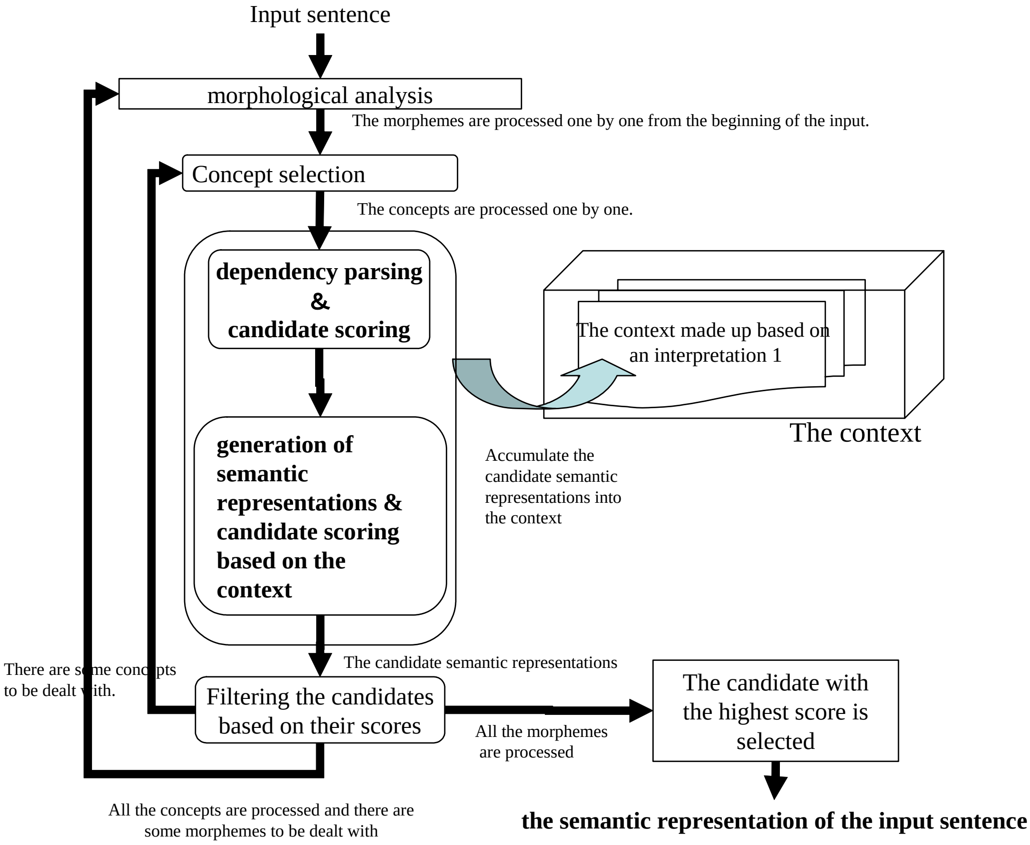

Input sentence
morphological analysis
The morphemes are processed one by one from the beginning of the input.
Concept selection
The concepts are processed one by one.
dependency parsing & candidate scoring
generation of semantic representations & candidate scoring based on the context
The context made up based on an interpretation 1
The context
Accumulate the candidate semantic representations into the context
The candidate semantic representations
There are some concepts to be dealt with.
Filtering the candidates based on their scores
All the morphemes are processed
The candidate with the highest score is selected
All the concepts are processed and there are some morphemes to be dealt with
the semantic representation of the input sentence

Figure7: Process flow

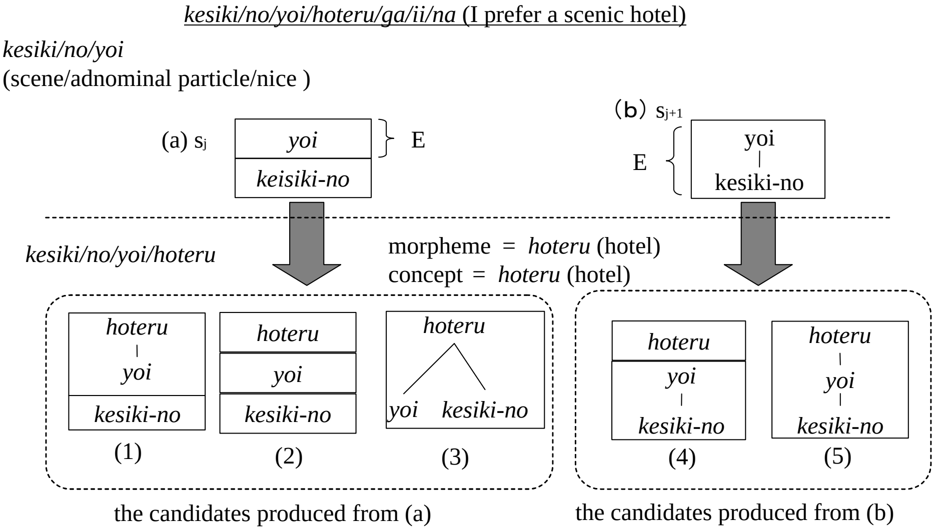


Figure8: Aanalytic process of the phrase, *kesikki-no-yoi-hotel*

3.2.Assignment of Scores

In scoring each candidate, we employ context-free criteria and context-dependent criteria in Table 2. The score of a candidate is given by adding the context-free score and the context-dependent score. The context-free criteria involve grammatical criteria, (s1-3) (e.g. a ban on long dependency), statistical criteria, (st0), and semantic criteria independent of the context, (c1-6) (e.g. selectional restriction between a verb and its arguments). We tentatively define a weight of each criterion as follows: context-dependent criteria (+3) > context-free semantic criteria (+2) > grammatical criteria and statistical criteria (±1). This is partly because we do not have a corpus big enough to determine a precise weight of each criterion. (The criterion, (g1), deduces a point (-1). The reason comes from the fact that the Japanese language generally prefers shorter dependency for longer one.)

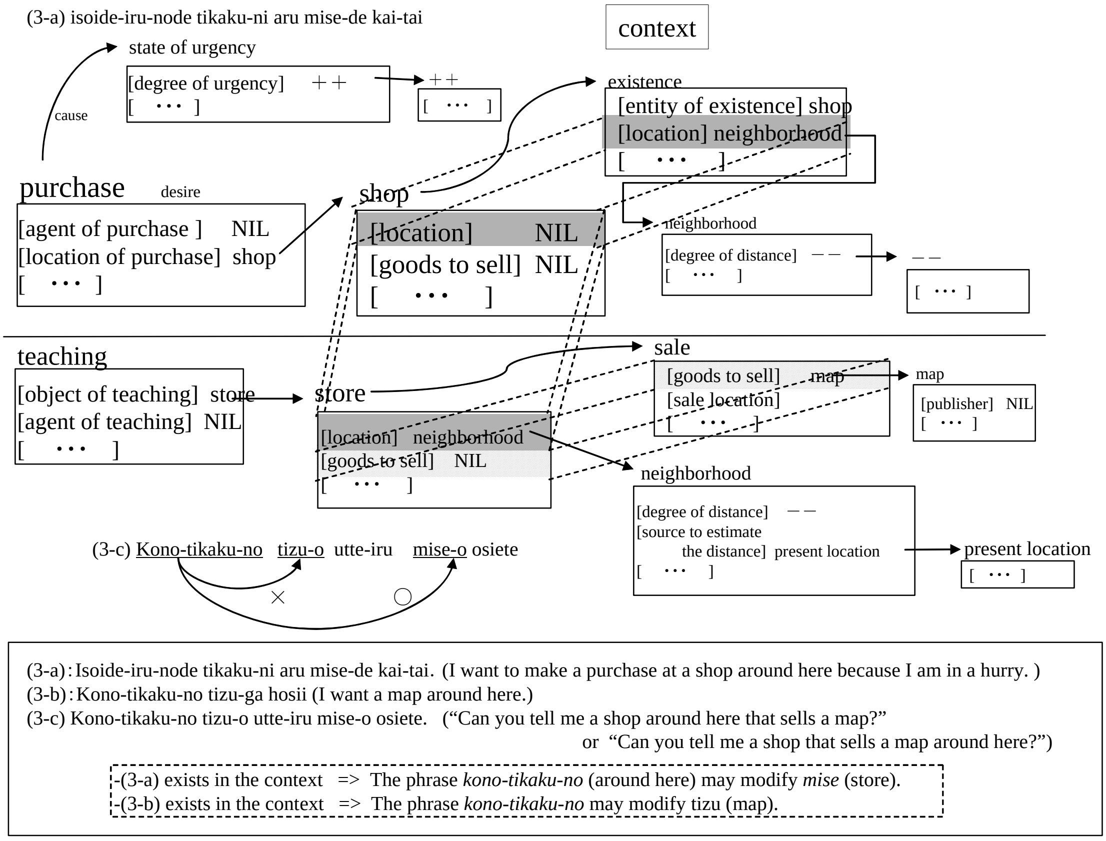
Many of the context-dependent criteria are based on whether an attribute-value pair is found in the context. The criterion (c1) raises a score when a generated attribute-value pair conforms to another attribute-value pair (i.e. when they are in a super-sub or whole-part relation. The criteria (c2) and (c3) are based on the relationship between attribute-pairs in a noun and attribute-value pairs in a predicate of an adnominal clause. The criterion (c2) raises a score when a pair in a noun conforms to another pair in a predicate of an adnominal clause. The criterion (c3) does the same operation when a noun concept could be regarded as the value of an attribute in the modifying predicate. The criterion (c4) raises a score when one of the attribute-pairs in question is a primary attribute and the other is a sub-attribute associated with it.

The criteria (c5) and (c6) disambiguate polysemic/homophonic words. When the system finds the same concept as a retrieved concept, it raises a relevant score. When the system receives a morpheme, it extracts every concept associated with the morpheme. (c5) and (c6) then plays a role if the morpheme is polysemic/homophonic. As for noun concepts, a score is also raised when the context contain another concept in the same semantic class as a retrieved noun concept.

Table 2: Scoring criteria

context-free criteria	g1	The dependency is longer than the shortest possible dependency. (This is not applied when a depending phrase is introduced by a topic particle *wa*.)	Deduce points
	g2	A linking word (bound morpheme) connecting two content words conforms to a linking word designated in the concept dictionary.	Add points
	g3	A concept denoting a unit has dependency with a concept denoting the value (e.g. when the concept “yen” has dependency with the concept “7,000”).	Add points
	st0	Dependency between given two words frequently appear in the corpus.	Add points
	s1	An attributes of an adjective/adverbial adjective matches with an attribute of the modified noun; an attribute of an adverb matches with an attribute of the modified words (i.e. verbs, adjectives, etc.).	Add points
	s2	Coordinated nominal concepts belong to the same semantic class.	Add points
	s3	Two words in dependency has a super-sub/whole-part relation.	Add points
context-dependent criteria	c1	A generated attribute-value pair matches with an attribute-value pair in the context.	Add points
	c2	An attribute-value pair in a clause modifying a noun concept matches with an attribute-value pair in the noun concept.	Add points
	c4	A noun concept modified by an adnominal clause can be interpreted as the value of an attribute of a predicate in the modifying clause.	Add points
	c4	A generated attribute-value pair and an attribute value pair in the context have the relation between a primary attribute and a sub-attribute.	Add points
	c5	A concept being processed has already appeared in the context.	Add points
	c6	The context has a proper name which belongs to a sub-class of a common noun being processed.	Add points

3.3.Example Operation



(3-a): Isoide-iru-node tikaku-ni aru mise-de kai-tai. (I want to make a purchase at a shop around here because I am in a hurry.)
(3-b): Kono-tikaku-no tizu-ga hosii (I want a map around here.)
(3-c) Kono-tikaku-no tizu-o utte-iru mise-o osiete. ("Can you tell me a shop around here that sells a map?"
or "Can you tell me a shop that sells a map around here?")

-(3-a) exists in the context => The phrase *kono-tikaku-no* (around here) may modify *mise* (store).
-(3-b) exists in the context => The phrase *kono-tikaku-no* may modify tizu (map).

Figure9: Disambiguation of modification relation

Figure 9 shows an example in which the system disambiguates possibly ambiguous dependency structure. The sentence (3-c) *Kono-tikaku-no tizu-o utte-iru mise-o osiete* is two-way ambiguous if we ignore the context. The phrase *kono-tikaku-no* (around here) may modify either *mise* (shop) or *tizu* (map). In the former case, the sentence would mean "Can you tell me a shop around here that sells a map?" and, in the latter case, it would mean "Can you tell me a shop that sells a map around here?" If (3-a) is contained in the context, the former should be the case. If (3-b) appears in the context, on the other hand, the latter should be the case. In either case, our context-dependent criteria give a higher score to a candidate which contains the same attribute-value pair as the one in the context. Figure 9 illustrates the former case, in which the relevant phrase should be interpreted as modifying the farther noun (i.e. *mise*).

4. Evaluation

4.1 Experimental design

We evaluated the proposed method by using dialogs about hotel search and reservation. The dialogs were collected from seven subjects, who are graduate and undergraduate students majoring in computer science. The experiment was designed as follows.

- Two of the subjects engaged in a conversation in which one played a role of a guest looking f or a hotel and the other played a role of an operator who actually makes hotel search/reservation for the guest.
- The conversations were made in a non-face-to-face environment; that is, the subjects could not use non-verbal information like gestures. There were no restrictions imposed on words and sentence styles used in the conversations.

- We prepared five different situations for hotel search/reservation, and randomly assigned one of them to a pair of the subjects. The pair were asked to make a conversation in which the guest looks for a hotel designated in the assigned situation.

Table 3:experimental data

Collected daialogs	24	
Utterances (Sentences)	Guest	667(746)
	Operator	723(904)
Subjects	7	
Sentences to be analyze	200	

Table 3 shows the number of collected utterances and the related figures. The evaluation was made based on the 667 utterances (746 sentences) made by the guests because the dialog system plays a role of an operator. We made some modifications to the 746 sentences. The modifications included rewriting extremely informal spoken style into written style, deletion of fumbling and/or rephrasing, rewriting of regional dialects into standard expressions, and so on. We further excluded sentences made up of less than three free morphemes. The excluded sentences include “Hai (Yes)”, “Onegai-shimasu (Do it, please)”, “Ee (Yah),” “So-desu (You are right)”, and so on. The remaining 200 sentences were examined based on the following two criteria.

(1) Did the system correctly parse input sentences?

(2) Did the system generate correct semantic representations?

When the system processed an input sentence, we manually gave the system correct semantic representations composing the associated context. We then manually checked whether the system generated correct dependency structure and semantic representations. The frame dictionary and the concept hierarchy used in the evaluation had been prepared before the experiment without referring to the experiment data.

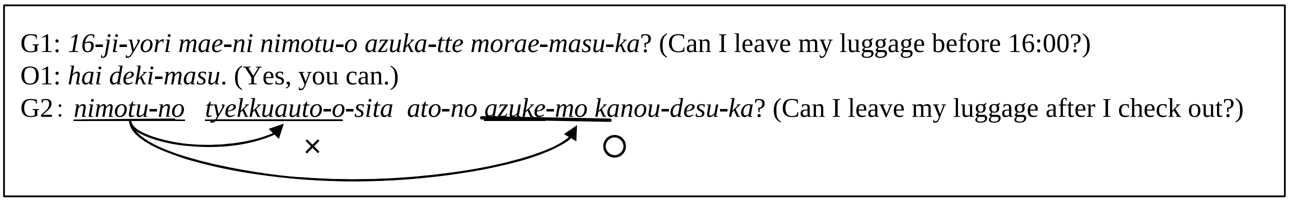
4.2 Discussion

Table 4: Accuracy of parsing (%)

	KNP	Cabocha	Proposed Method
Without context	83	82	85
With context	-	-	87

Table 5: Accuracy of semantic analysis (%)

Without context	57
With context	64



G1: *16-ji-yori mae-ni nimotu-o azuka-tte morae-masu-ka*? (Can I leave my luggage before 16:00?)

O1: *hai deki-masu*. (Yes, you can.)

G2: *nimotu-no tyekkuauto-o-sita ato-no azuke-mo kanou-desu-ka*? (Can I leave my luggage after I check out?)

Figure10: Determination of longer dependency based on the context.

Table 4 shows the result of parsing. When the system was given the relevant context, the accuracy increased by about 2 % compared to the parsing result without the context. The result confirms the effectiveness of using the context in parsing. Table 4 also shows the result of parsing by two other parsers which do not refer to the context, for comparison. Our system achieved a higher accuracy than the two parsers because out system has components (the concept dictionary, etc.) made specifically for hotel search/reservation. The contextual information raised the accuracy because it helped the system to assign a higher score to a candidate containing a longer dependency than to a candidate containing a shorter dependency. Table 4 indicates the accuracy of the system where the system assigned the highest score to a unique candidate and the candidate had the correct dependency structure. Figure 10 shows an example involving a longer dependency that the system successfully obtained. In this example, the context contains an attribute-value pair "object to be left = baggage" and the system gives a higher score to the long dependency based on the attribute-value pair in the context.

Table 5 shows the accuracy where the system gave the highest score to a unique candidate, which was made up of correct semantic representations. When the system did not refer to the context, the accuracy was 57% (114/200). Reference to the context raised the score to 64% (128/200). It confirms that reference to the context is effective not only in parsing but also in generating correct semantic representations. Reference to the context helps the system to assign a higher score to a candidate involving correct semantic representation. Take a phrase "ikura-no hoteru (literally "how much hotel") for example. Since the concept *hoteru* has more than one attribute associate with rate (e.g. hotel rate, parking rate, etc.), the system generates attribute-value pairs like "hotel rate = ?rate" and "parking rate = ?rate". By referring to the context, the system obtains the correct semantic representation which involves the correct attribute-value pair that also appears in the context.

The system failed in generating correct dependency structure and/or semantic representations in some examples even when it referred to the context. One typical example involves meta-utterances referring to action took in the conversation.

(U3) Basu-no jikoku-hyo-o sakihodo itta denwa-bango-e okutte-kudasai.
(Please send the bus schedule to the phone number mentioned previously.)

In order to generate the candidate in which *jikoku-hyo-o* (bus schedule) has dependency with *okuru* (send), it is necessary to identify the subject/agent of *itta* (mentioned) and to understand who said what at which phase in the conversation. Another typical example does not require the context for disambiguation but the disambiguation requires more knowledge than our system currently has.

(U4) X hoteru-no kuruma-de iku iki-kata-o oshie-te
(Tell me how to get to X Hotel by car.)

In (U4), *X hoteru* (X hotel) has dependency not with *kuruma* (car) but wih *iki-kata* (how to get to). In order to obtain the correct dependency, the system must have knowledge that customers may go to a hotel by car, bus, taxi, etc. This kind of ambiguity would be solved by describing relevant knowledge in the concept dictionary in the same form as the context.

5. Conclusion

We proposed the framework to analyze sentences by integrating dependency analysis with semantic analysis referring to the context. This method sequentially generates semantic representations and compares attribute-value pairs in input and the context. It has been confirmed that the method is helpful to resolve syntactic/semantic ambiguity.

The most significant feature of this method is that it does not require complicate inference rules in using context information. This is achieved by making up the context with stable semantic representations. The system needs a framework for keeping more than one interpretation and making a natural response because not every input can be disambiguated. When the context information is not sufficient to disambiguate an input, the system needs to use knowledge about the entities/events mentioned or knowledge about the domain of the dialog. How to use such knowledge remains one of the issues in the future research.

6. References

Charniak, E. 2000. A Maximum-Entropy-Inspired Parser, *Proceedings 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 132–139.

Harada, M., and Mizuno, T. 2000. Japanese Semantic Analysis System SAGE using EDR. *The Japanese Society for Artificial Intelligence*, Vol. 16, No. 1, pp. 85–93, 2000(in Japanese)

Kaplan, R. M., and Bresnan, J. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. *The Mental Representation of Grammatical Relations (Joan Bresnan, ed.), The MIT Press,* pp.173-281.

Kurohashi, S. 1998. Japanese Dependency/Case Struc-ture Analyzer KNP version 2.0b6. Department of Infor-matics, Kyoto University. (in Japanese).

Kudo, T., and Matsumoto, Y. 2002. Japanese Dependency Analysis using Cascaded Chunking. *Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*,pp.63-69.

Noguchi, Y., Ikegaya, Y., Takagi, A., Nakashima, H., Konishi, T., Itoh, T., Kondo, M., and Itoh, Y. 2002. A Framework for Semantic Representations for a Natural Language Dialog System, *Proceeding of SNLP-Oriental COCOSDA*, pp.231-236.

Pollard, C., and Sag, I. A. 1994. Head-Driven Phrase Structure Grammar. *University of Chicago Press*,.

Takagi, A. and Y. Itoh. 1987. "Natural Language Processing ", Maruzen, (in Japanese).

Yarowsky, D. 2000. SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguating Systems., *The Association for Computational Linguistics*.