Towards Japanese Ellipsis Resolution with a Generalized Discrimination Network

Manabu Okumura, Hozumi Tanaka
Department of Computer Science,
Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku, Tokyo, 152, Japan
Tel: (+81-3)3726-1111 ext. 4188

Fax: (+81-3)3720-4925 email: dora@cs.titech.ac.jp

Topic: Natural Language(discourse)

Category: long paper

Keywords: ellipsis resolution, discrimination network, case frame, abduc-

tion, coherence, commonsense knowledge

Abstract

In this paper, we present a framework for ellipsis resolution that aims at the cooperative integration of syntactic, semantic, and pragmatic strategies. Our framework is based on a generalized discrimination network(GDN), which we proposed for incremental word sense disambiguation. GDN is a variant of a discrimination network that can solve a big problem of a discrimination network that it can only be traversed in an a priori-fixed order of obtained constraints. GDN-represented case frames easily execute the detection of the places of gaps, while GDN-represented

commonsense knowledge, together with GDN-represented case frames, execute coherence-directed inference, which enables the filling of the gaps as a by-product. Our approach is based on Halliday and Hasan's notion of cohesion, and reflects our view that overtly underspecified types of cohesion, such as ellipses and pronominal anaphora, have less information to resolve in themselves and are handled by coherence-directed inference in terms of overtly specified types of cohesion, such as lexical cohesion and conjunction.

1 Introduction

Ellipses occur frequently in Japanese sentences whenever a hearer(reader) can easily predict the gaps and understand the sentences mainly through the contextual information. And so the ellipsis resolution process is indispensable for Japanese understanding systems.

The ellipsis resolution process can be divided into two phases: detection of the place of gaps in a sentence, and filling the gaps. Detection of the place of gaps has often been accomplished using case frames[5, 24] on the basis that all obligatory slots should be satisfied by phrases in a sentence, and so unsatisfied slots must be gaps. Then, semantic constraints are imposed on gaps because they are supposed to satisfy slots of a frame that have their own selectional restrictions. In the phase of filling the gaps the most appropriate phrase is found from the contextual information that satisfies the semantic constraints.

Ellipsis resolution, especially the phase of filling the gaps, might be influenced by a combination of syntactic, semantic, and pragmatic knowledge. Previous researches on ellipsis resolution, however, have emphasized syntactic[26] and semantic strategies[5] or have concentrated on using pragmatic knowledge[4]. In this paper, we present a framework for ellipsis resolution that aims at the cooperative integration of syntactic, semantic, and pragmatic strategies¹.

Our framework is based on a generalized discrimination network(GDN), which we proposed for incremental word sense disambiguation[18]. GDN is a variant of a discrimination network[7] that can solve a big problem of a discrimination network that it can only be traversed in an a priori-fixed order of obtained constraints.

First, we show that as a representation form of a set of case frames, GDN can easily detect which phrases are omitted, and can check the possible substitutions for the missing phrases by means of the selectional restrictions (detection phase).

Next, we present an inference directed by 'coherence,' which makes a relation between two sentences based on explicit information in the sentences and background knowledge, and fills the gaps in the sentences as a by-product of establishing coherence. Our approach is based on Halliday and Hasan's

¹We treat only ellipses of postpositional phrases.

notion of cohesion[9], and reflects our view that overtly underspecified types of cohesion, such as ellipses and pronominal anaphora, have less information to resolve in themselves and are handled by coherence-directed inference in terms of overtly specified types of cohesion, such as lexical cohesion and conjunction. We show that as a representation form of both a set of case frames and a set of rules(commonsense knowledge), GDN is suitable for the coherence-directed inference mechanism(filling phase).

In section two, GDN is outlined. In section three, we describe how GDN can detect the place of gaps. In section four, we first present a coherence-directed inference, and then show merits of GDN as a coherence-directed inference mechanism. Finally, we describe how GDN can fill the gaps as a by-product of coherence-directed inference.

2 Outline of generalized discrimination networks

In this section, we outline the principles of GDN to make comprehensible the explanation of the mechanism for ellipsis resolution using GDN, described in sections three and four. For formal details, please refer to [18].

2.1 Characteristics of discrimination networks

A discrimination network is a generalization of a decision tree[3] and has been used for various problem solving systems[7], especially in natural language processing to represent multiple word meanings compactly[11, 15, 13, 1]. A discrimination network is considered to be a directed acyclic graph with one root node and many leaf nodes. Figure 1 is a sample discrimination network². Each branch of the network has a constraint as its label. Each leaf node of the network points to a solution. Other nodes represent a set of possible solutions that correspond to leaf nodes below them, because further traversal along branches to multiple nodes are possible from them. The root node corresponds to the set of all solutions.

The problem solving process using discrimination networks is a step by step downward traversal of the network from the root node to a leaf node,

²For expository clarity, we use only tree-form examples in this paper.

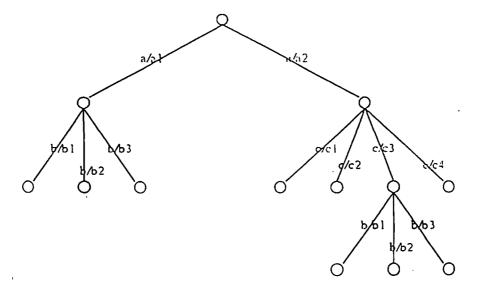


Figure 1: A sample discrimination network

guided by branches which satisfy the obtained constraints. In this process, inappropriate alternatives are rejected and appropriate solutions are selected. Reaching a leaf node means that a solution is found.

A discrimination network has the following advantages:

- in the discrimination network, a unique node can represent multiple solutions that correspond to the leaf nodes below it. Therefore, the downward traversal of the network corresponds to the continuous refinement of a solution into a more specific one. This is what the constraint programming paradigm[8] will achieve;
- compared with a linear search, the discrimination network's search algorithm is more efficient because the downward traversal is guided by constraints that are labels of branches, and the search space can be gradually narrowed down. This search takes time O(l) in the worst case, where $l = \log n$ is the height of the network, while the linear search takes time O(n) in the worst case, where n is the number of all possible solutions[2];
- we can compact a set of rules, because some of the same preconditions(constraints) can be merged into one branch in the network. Therefore, the amount of checking on whether a constraint is satisfied is reduced, and wasteful repeated computations can be avoided.

Although a discrimination network has the excellent characteristics mentioned above, it has two critical problems. The first one is that it cannot be traversed unless constraints are entered in an a priori-fixed order. Because the order in which constraints are obtained cannot be a priori fixed in general, the process of traversing the network downward may often have to be suspended until the constraint in the right order is obtained. Because the network is traversed downward from the root node, constraints must be entered one by one from constraints that are labels of branches connected to the root node. The second problem is more serious. If some constraints, which are necessary for traversing the network downward, are not obtained, the problem solving process will not be able to traverse the network, and a deadlock will occur. In this situation, the constraints that have already been obtained will cease to contribute to the problem solving process.

To solve the above problems, we proposed a 'generalized discrimination network(GDN).' The problem solving process can traverse the GDN immediately whenever any constraints are obtained.

2.2 Principles of generalized discrimination networks

Consider the discrimination network shown in Figure 1. To represent the network as a table, we make stages of preparation. First, a numerical string is assigned to each node as a unique identifier. To the nodes in Figure 1, identifiers are assigned, as shown in Figure 2.

Second, constraint-identifier pairs are extracted from the network in the following form: a branch, and a subordinate node that is directly connected by the branch. This correspondence between constraint and identifier means that if a constraint is satisfied, the nodes of corresponding identifiers can be reached in the network. For example, if constraint c/c_2 is satisfied, the network can be traversed downward to the node of corresponding identifier 122.

Here, we must pay attention to the other constraints in the path from the root node to the reached nodes. In the case of the node of identifier 122 mentioned above, constraint a/a_2 exists and is unsatisfied. Therefore, the reachability of node 122 is 'conditional,' because node 122 can be reached if constraint a/a_2 is satisfied.

Hence, from Figure 2, pairs of constraints and 'conditional identifiers,' that is, identifiers of the conditionally reached node, are obtained in Table

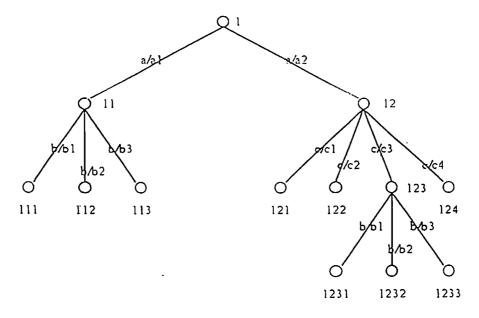


Figure 2: Discrimination network with identifier-assigned nodes

1. A conditional identifier consists of an identifier of the reached node, followed by an 'if-clause' which represents a list of unsatisfied constraints. An identifier with no if-clause means that a node of the identifier can be reached unconditionally.

The preparation is finished. The regular order of constraints is a/a_2 , c/c_3 , b/b_1 for traversal of the network in Figure 2 downward to node 1231. Here, in contrast, the case in which constraints are obtained in the order of b/b_1 , c/c_3 is considered. Notice that in this case, a necessary constraint a/a_2 is not obtained and that the order of the obtained constraints is irregular. The discrimination process in our approach for that case is briefly described below. Figure 3 shows a 'state' transition that represents the discrimination process. A state is in the form of a conditional identifier. The initial state(a state in which no constraints are obtained) is 1 with no if-clause(the identifier of the root node). Informally, after constraint b/b_1 is obtained, the state is computed as follows, with the current state(the initial state) and a set of conditional identifiers $\{111 \text{ if } a/a_1, 1231 \text{ if } a/a_2 \text{ and } c/c_3\}$ corresponding to the obtained constraint by Table 1:

Both 111 and 1231 include 1 as a prefix-numerical string, so the longer strings 111 and 1231 are returned. Because the current state has no if-clause, the if-clause of the next state becomes the same as the if-clause of the conditional identifiers cor-

```
a/a_1 11

a/a_2 12

b/b_1 {111 if a/a_1, 1231 if a/a_2 and c/c_3}

b/b_2 {112 if a/a_1, 1232 if a/a_2 and c/c_3}

b/b_3 {113 if a/a_1, 1233 if a/a_2 and c/c_3}

c/c_1 121 if a/a_2

c/c_2 122 if a/a_2

c/c_3 123 if a/a_2

c/c_4 124 if a/a_2
```

Table 1: Correspondence between constraint and node identifier

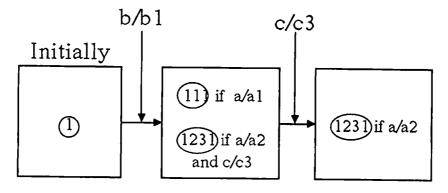


Figure 3: The discrimination process using GDN

responding to the obtained constraint. The if-clause of the obtained constraint represents a list of constraints between the root node(1) and the reached node($\{111, 1231\}$) except the obtained constraint(that is, if a/a_1 , if a/a_2 and c/c_3) respectively. Therefore, the next state becomes $\{111 \ if \ a/a_1, 1231 \ if \ a/a_2 \ and \ c/c_3\}$.

As shown in Figure 2, identifiers of mutually reachable nodes in the network are in a prefix-numerical string relation with each other. Therefore, the operation between identifiers can easily check whether one node can be reached from the other in the network. If one node is reachable from the other, the identifier of the subordinate one is returned. This operation cor-

responds to a downward traversal of the network by satisfying the obtained constraints. The analysis will fail if one identifier is not a prefix of the other.

Next, constraint c/c_3 corresponds to conditional identifier 123 if a/a_2 . When the constraint is obtained, the state has multiple conditional identifiers {111 if a/a_1 , 1231 if a/a_2 and c/c_3 }, and so operations are performed on each conditional identifier with the constraint:

As for $111 \ if \ a/a_1$, the identifiers are not in a prefix relation, so the analysis fails. Therefore, the result is necessary only for conditional identifier $1231 \ if \ a/a_2 \ and \ c/c_3$. The resultant identifier is 1231 from identifiers 123 and 1231. The resultant if-clause is $if \ a/a_2$, because constraint c/c_3 in the if-clause of the current state is obtained and removed from it. Hence, the final result of traversal for the above case becomes $1231 \ if \ a/a_2$, which means that node 1231 is reachable if constraint a/a_2 is obtained.

The if-clause allows us to cope with the irregular order of the obtained constraints and the lack of necessary constraints. The if-clause is the storage of constraints that must be obtained to reach the destination node but have not been obtained yet. When constraints are obtained in an irregular order, the constraints in the wrong order can be found in the if-clause, and are removed from it.

Thus, constraints in the if-clause are expected to be obtained, and they can be used as predictions for constraints that will be obtained later. From another viewpoint, constraints in the if-clause can be regarded as assumptions in the abductive reasoning[10, 17], because the if-clause means that the conclusion of a rule(corresponding to the leaf node) holds, and so preconditions(constraints) in the if-clause must hold in the future.

3 Detection of the place of gaps using GDN

Figure 4 is a portion of the discrimination network that represents the word senses (and case frames) of the Japanese verb 'naosu.' Each branch of the network has as its label a selectional restriction on surface cases, such as postpositions 'ga,' 'wo,' and so on. Each leaf node of the network points to a unique word sense, which is represented by the underlined label. A set of

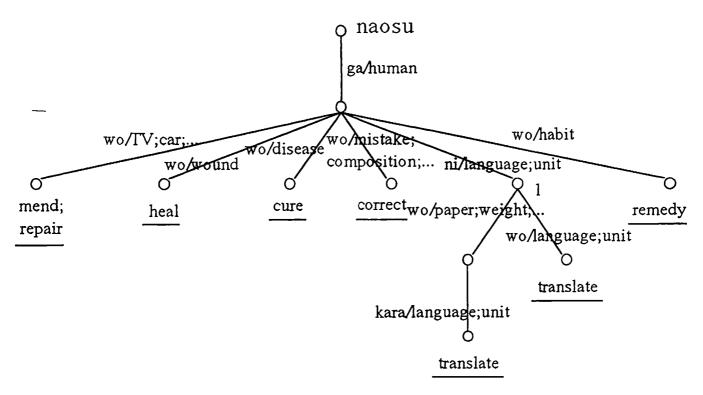


Figure 4: A portion of the discrimination network of the word senses of the verb 'naosu'

constraints in the path from the root node to a leaf node represents the case frame for the word sense corresponding to the leaf node.

Works such as [11, 15, 13, 1] realize the word sense disambiguation process as a downward traversal of the discrimination network. In these works, the merits of such a network are described, as mentioned in section 2.1.

Using GDN instead of a discrimination network adds another advantage, in that the word sense disambiguation process proceeds incrementally[18]. We think incremental disambiguation[14] is a better strategy for word sense disambiguation, because a combinatorial explosion of the number of ambiguities might occur unless word sense ambiguity is incrementally resolved as early as possible whenever constraints are obtained during the analytical process of a sentence.

Here, we describe how GDN deals with ellipsis detection, which finds the place of missing phrases. Because GDN is a variant representation form of a set of case frames, the ellipsis detection approach with GDN is case frame-based[5, 24]. Nonetheless, it can easily detect which phrases are omitted, and can check the possible substitutions for the missing phrases by means of the selectional restriction, as described below.

The positions of the missing phrases are detected in two ways with GDN. First, they are explicitly indicated by the if-clause mentioned in section 2.2 because constraints in the if-clause are not obtained but are expected. Secondly, in the case in which the reached node is not a leaf node when the analysis of the whole sentence is finished, constraints between the current node and leaf nodes are considered to be the positions of the missing phrases, because the verb should have a unique word sense³. So in such a case, ellipsis resolution enables the further disambiguation of the word sense. By applying to the candidates for the substitution the selectional restriction that corresponds to the detected position of the missing phrase, the semantic appropriateness of the substitution is checked, and all candidates that violate it are eliminated.

Now we illustrate how incremental word sense disambiguation and ellipsis resolution proceed jointly in Japanese. Consider the sentences:

```
taro ga nihongo de ronbun wo kaki, (Taro Japanese a paper wrote) eigo ni hanako ga \phi naosita. (EnglishHanako it)
```

Taro wrote a paper in Japanese, and Hanako translated it into English.

Here, ' ϕ ' indicates the gap. In the example, the word sense of the verb 'naosu' is ambiguous, as shown in Figure 4⁴, and the order of obtained constraints 'ni/eigo(English), ga/hanako(Hanako)' is irregular compared with the network's order. Constraint 'ga/human' must be obtained earlier than 'ni/language; unit' in the network. Additionally, the phrase corresponding to 'it' is omitted.

The second sentence is analyzed, and the place of gaps is detected as follows:

'Eigo'(English) is a language and satisfies a selectional restriction of a postposition 'ni,' and node 1 is reached. The word sense

³Of course, another mechanism for checking the syntactic grammaticality is necessary in the case of languages such as English, because it is strange to say that 'I repair.' is grammatical, and the object is omitted in the sentence. As for Japanese, such a mechanism seems unnecessary.

⁴Only the correct word sense, however, is written in the English translation.

ambiguity of the verb 'naosu' has been reduced from seven to two, but the word sense is still ambiguous⁵. Next, 'hanako'(Hanako) satisfies a selectional restriction of 'ga,' and the reachability of node 1 becomes unconditional. After the verb 'naosu' is analyzed, the further traversal is attempted from node 1, because it does not point to a unique word sense. As a result, we get the candidate positions of the missing phrases, that is, {'wo/language;unit'}, {'wo/paper;weight', 'kara/language;unit'}.

Then, assuming the candidates for filling the gaps are ['ronbun'(a paper), 'nihongo'(Japanese), 'taro'(Taro)] from the first sentence, where the first element is preferred⁶, ellipsis resolution is completed as follows:

Because 'ronbun' (a paper) is a paper and 'nihongo' (Japanese) is a language, they satisfy a selectional restriction of postpositions 'wo' and 'kara,' respectively. Thus, the substitutions are decided and the word sense of the verb 'naosu' is uniquely determined. The final result for the second sentence is 'Hanako translated the paper (which Taro wrote) from Japanese into English.'

In the next section, we explain the real mechanism for filling the gaps using GDN.

4 Filling the gaps using GDN

4.1 Knowledge for filling the gaps

Ellipsis resolution, especially the phase of filling the gaps, might be influenced by a combination of knowledge sources. Other than semantic constraints (selectional restrictions) for slots of frames, which help find an appropriate phrase for gaps, syntactic and semantic constraints, such as coincidence of surface or deep cases in main and subordinate sentences, or in a sentence and the preceding sentence, have been used to fill the gaps [26, 5].

⁵We think there is a subtle difference between the remaining two word senses because they have different case frames.

⁶Here, the naive heuristic 'recency' is assumed to find candidates, and to decide the order of priority among them.

As for the example in section three, candidate positions of the missing phrases are {'wo/language;unit'}, {'wo/paper;weight','kara/language;unit'}. In these, the position of case 'wo' is filled according to the above constraints as follows:

The phrase with case 'wo' in the preceding sentence, that is 'ronbun' (a paper), satisfies a selectional restriction of case 'wo' of the missing phrase. Thus, the substitution is decided in that the missing phrase of case 'wo' in the second sentence is filled by the phrase 'ronbun' in the first.

Indeed, some examples are handled, but many examples of ellipses cannot be handled by syntactic and semantic constraints. Consider the following example:

```
taro ga hanako ni denwabangou wo kiitaga, (Taro Hanako phone number asked) \phi_1 \phi_2 \phi_3 osietekurenakatta. (she it him did not tell)
```

Taro asked Hanako her phone number, but she did not tell it to him.

As for this example, the above constraints yield an incorrect substitution in that Taro did not tell Hanako his phone number, because phrases of case 'ga', 'ni', and 'wo' in the first sentence fill the missing phrases of the same case in the second, respectively.

To handle such examples problematic for syntactic and semantic strategies, pragmatic knowledge, such as speaker's plans and goals, has been used[4].

As for the above example, assuming commonsense knowledge, such as 'if someone is asked something by some other person, then s/he will tell it to the person,' the missing phrases are correctly filled in that Hanako(who was asked) is the subject of 'tell' and Taro(who asked) is the indirect object of 'tell.'

One problem of previous approaches to ellipsis resolution is that they have provided only a mechanism to process a particular kind of knowledge source, yet they do not insist that only the kind of knowledge source should be used. In the following sections, we present a unified approach to ellipsis resolution

that aims to utilize all of the knowledge sources mentioned above, and fill the gaps uniformly using a single coherence-directed inference mechanism with GDN.

4.2 Coherence-directed inference

In context analysis, it is indispensable to make connections between two sentences which are implicit on the surface. If it is possible to make connections between them, they are said to be 'coherent.' We regard the task to make connections between sentences as primary and other tasks in context analysis, such as ellipsis and anaphora resolution, as secondary. This is because more useful information is obtained from surface sentences for making sentences coherent than for ellipsis and anaphora resolution alone. Thus, we adopt an approach for ellipsis resolution in which filling the gaps is executed as a by-product of making connections of two sentences.

To make connections between two sentences, inference is necessary based on both explicit information in them and world knowledge. First, we indicate what explicit information for inference is available in the sentences. Halliday and Hasan[9] regard the notion of cohesion as a clue that is available in surface sentences, and that is useful in making sentences coherent. Cohesion is divided into the following five categories:

- 1. Reference
- 2. Substitution
- 3. Ellipsis
- 4. Conjunction
- 5. Lexical Cohesion

Of these, we use only Conjunction and Lexical Cohesion, as they are overtly specified in sentences and are useful in making connections between events in the sentences. The first three of them, however, have less information in surface sentences to make an inference. For example, the pronoun

'he' provides only the information of referring to a male, and what is worse, a missing phrase provides no information⁷!

Structural or semantic parallelism in Conjunction is a useful clue in making sentences coherent. We can infer that two sentences are coherent if they are located in mutually parallel positions. Consider the following example:

```
taro ga hanako ni hana wo age, (Taro Hanako flower gave) jiro ga \phi hon wo okutta. (Jiro to her book presented)
```

Taro gave Hanako a flower, and Jiro presented a book to her.

These two sentences in the example seem coherent because they have similar constructions and consist of similar event structures (case frames), that is, their verbs and phrases of mutually corresponding case slots are respectively synonymous or antonymous. In this case, gaps in one sentence can be filled by the phrases of the corresponding case slots of the other sentence.

Lexical Cohesion is concerned with connecting events by means of inference on commonsense knowledge. Briefly, we regard two events as coherent if there exists an event that is a common precondition or conclusion of them in a set of rules (commonsense knowledge)⁸. Consider the following example:

```
taro ga kabin wo hakondeitaga, (Taro vase was carrying) \phi_1 \phi_2 otositesimatta. (he it dropped)
```

Taro was carrying a vase, but he dropped it.

We guess that two sentences are coherent because both the events 'carry' and 'drop' have event 'hold' as a precondition. That is, we assume the following rules as the basis of coherence of the two sentences.

$$hold(X,Y) \land move(X) \rightarrow carry(X,Y).$$

⁷Information of focus(centering)[22, 12, 25] is considered useful information for anaphora and ellipsis resolution available in the context.

⁸Those cases where one event is a precondition or conclusion of the other, such as the example in section 4.1, are also included in this category.

$$hold(X,Y) \wedge let_{-}fall(X,Y) \rightarrow drop(X,Y).$$

In this case, gaps can be filled by the phrases that share the same argument variable in the rules used for making sentences coherent.

The result of inference is more preferable (two sentences are more coherent) if more information for coherence is obtained through inference, such as similarities or common preconditions of two events. Thus, inference is guided towards making two sentences more coherent in the above mentioned sense.

In our approach, filling the gaps in ellipsis resolution is executed as a by-product of the general process of establishing coherence. Prüst and van der Berg[19] adopt the similar approach to ellipsis resolution, and assume that for two sentences to be coherent, semantic structures of them should have something in common related to conceptual hierarchy. Their approach, however, can treat only cases of structural or semantic parallelism. We extend the notion of commonness, and can treat cases where two sentences have something in common related to general commonsense knowledge.

Ng and Mooney propose similar metric 'explanatory coherence' [16] for abduction applied to natural language processing, and show that it plays a role in guiding the inference towards plausible solutions. But we think their metric is weaker for the following reasons:

- It does not truly reflect the linguistic notion of 'coherence.' Their notion of coherence is based on the number of common preconditions of two observations(events). There is a danger, however, that two observations in a sentence might be considered, while two observations in different sentences must be considered to establish coherence between sentences. It is because the notion of 'sentence' is not taken into account;
- It uses only Lexical Cohesion a la Halliday and Hasan's notion of cohesion. What is worse, it cannot handle cases as follows, in which two sentences are coherent because the same fact is inferrable(predictable) from both of them.

John was lost. He pulled over to a farmer by the side of the road. (He asked him where he was.) Coherence metric should be applied not only to abduction, where preconditions are assumed to hold because conclusions hold, but to cases when conclusions are assumed to hold because preconditions hold, which is called default reasoning[20].

4.3 GDN as a coherence-directed inference mechanism

GDN has good characteristics, as mentioned in section 2.1. In addition to those, as a coherence-directed inference mechanism, GDN has the following merits:

- Similarities of two events' case frames(each having some common slots) are easily detected by GDN, because GDN represents a common case slot of some different frames as one branch of the network;
- As described in section 2.2, GDN naturally executes abductive inference, because constraints in an if-clause can be regarded as assumptions in the abductive reasoning[10, 17];
- As is clear from the third characteristic mentioned in section 2.1, GDN can easily detect common preconditions that appear in multiple rules. This is important because coherence metric is based on the detection of such common preconditions, which is also used as 'unify'[10] and 'factoring'[23] in many abductive systems. Similarly, GDN can also easily detect multiple rules that share a common conclusion.

4.4 Filling the gaps by coherence-directed inference

In this section, we illustrate how coherence-directed inference with GDN fills the gaps, using examples in section 4.2.

In the first example, a gap exists in the second sentence.

```
taro ga hanako ni hana wo age,

(Taro Hanako flower gave)

jiro ga \phi hon wo okutta.

(Jiro to her book presented)
```

Taro gave Hanako a flower, and Jiro presented a book to her.

Assuming the following case frames for the verbs 'ageru' and 'okuru,'9

```
ga/human, wo/thing, ni/human, ageru
ga/human, wo/thing, ni/human, okuru
```

the gap is represented as a constraint in the if-clause 'if ni/human.' So far as the above case frames are concerned, they are similar in that they share three case slots for 'ga', 'wo', and 'ni,' and so the two sentences are inferred as coherent. The shared case slots are merged into one branch in GDN, and because the case slot for 'ni' is shared and is satisfied by the phrase 'hanako' in the first sentence, the gap is also filled by 'hanako.'

In the second example, two gaps exist in the second sentence.

```
taro ga kabin wo hakondeitaga, (Taro vase was carrying) \phi_1 \phi_2 otositesimatta. (he it dropped)
```

Taro was carrying a vase, but he dropped it.

Assuming the following case frames for the verbs 'hakobu' and 'otosu,'

```
ga/human, wo/thing, hakobu
ga/human, wo/thing, otosu
```

the gap is represented as a constraint in the if-clause 'if ga/human and wo/thing.' In this case, the two sentences are inferred as coherent partly because the two case frames are similar. Moreover, assuming the following rules,

$$hold(X,Y) \land move(X) \rightarrow carry(X,Y).$$

 $hold(X,Y) \land let_fall(X,Y) \rightarrow drop(X,Y).$

both events 'carry' and 'drop,' which correspond to 'hakobu' and 'otosu' respectively, have event 'hold' as a common precondition, and so the two sentences are also inferred as coherent from this viewpoint. The arguments

⁹In this explanation, word sense ambiguity of verbs is ignored and all aspectual information is omitted.

X and Y of events 'carry' and 'drop' correspond to the phrases for cases 'ga' and 'wo' in the above case frames, respectively, and share the same value because two rules are merged through event 'hold.' As a result, the gaps for cases 'ga' and 'wo' are filled by 'taro' and 'kabin,' respectively, because both viewpoints support the explanation.

In the above examples, the application of the selectional restrictions to the possible substitutions for the missing phrases is unnecessary, because a common case slot of some different frames are merged into one branch, and the satisfiability of the selectional restrictions is not checked twice in GDN.

5 Conclusion

We described a framework for ellipsis resolution that uses GDN uniformly. GDN-represented case frames execute the detection of the places of gaps, while GDN-represented commonsense knowledge, together with GDN-represented case frames, execute coherence-directed inference, which enables the filling of the gaps as a by-product.

Our framework is imperfect, of course, in that it only employs a portion of useful knowledge sources; it should be augmented by other useful knowledge sources, and thus be able to treat more examples. Other factors, such as the information of focus(centering)[22, 12, 25] and clue words[21], must play a role too. The focusing rules might be applied to all candidates for the substitution, and contribute to a reduction in the number of surviving candidates before commonsense inference is carried out[22, 6]. Clue words, such as connectives, might help reduce the search space of coherence-directed inference, because connectives that mean 'explanation,' such as 'because,' indicate that the second sentence causes the first, which makes the direction of inference clearer. Determining the relative importance of these factors appears to be an interesting topic. It is our hope, however, that our framework will bring us closer to an integration of multiple strategies for ellipsis resolution.

References

- [1] G. Adriaens and S.L. Small. Word expert parsing revisited in a cognitive science perspective. In S.L. Small, G.W. Cottrell, and M.K. Tanenhaus, editors, Lexical Ambiguity Resolution: Perspectives from Psycholinguitics, Neuropsychology, and Artificial Intelligence, pages 13-43. Morgan Kaufmann Publishers, 1988.
- [2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. Data Structures and Algorithms. Addison-Wesley, 1983.
- [3] G. Brassard and P. Bratley. Algorithmics. Prentice Hall, 1988.
- [4] S. Carberry. A pragmatic-based approach to ellipsis resolution. Computational Linguistics, 15(2):75-96, 1989.
- [5] J.G. Carbonell. Discourse pragmatics and ellipsis resolution in taskoriented natural language interfaces. In *Proc. of the 21st Annual Meeting* of the Association for Computational Linguistics, pages 164-168, 1983.
- [6] D.M. Carter. Control issues in anaphora resolution. Report CRC-016, SRI International, 1990.
- [7] E. Charniak, C.K. Riesbeck, and D.V. McDermott. Artificial Intelligence Programming. Lawrence Erlbaum Associates, 1980.
- [8] M. Dincbas. Constraints, logic programming and deductive databases. In Proc. of the France-Japan Artificial Intelligence and Computer Science Symposium 86, pages 1-27, 1986.
- [9] H. A. K. Halliday and R. Hassan. Cohesion in English. Longman, 1976.
- [10] J.R. Hobbs and M. Stickel. Interpretation as abduction. In *Proc. of the* 26th Annual Meeting of the Association for Computational Linguistics, pages 95-103, 1988.
- [11] P.S. Jacobs. Concretion: Assumption-based understanding. In *Proc. of the 12th International Conference on Computational Linguistics*, pages 270-274, 1988.

- [12] M. Kameyama. A property-sharing constraint in centering. In *Proc. of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 200-206, 1986.
- [13] S.L. Lytinen. Are vague words ambiguous? In S.L. Small, G.W. Cottrell, and M.K. Tanenhaus, editors, Lexical Ambiguity Resolution: Perspectives from Psycholinguitics, Neuropsychology, and Artificial Intelligence, pages 109-128. Morgan Kaufmann Publishers, 1988.
- [14] C.S. Mellish. Computer Interpretation of Natural Language Descriptions. Ellis Horwood, 1985.
- [15] G.D. Moerdler and K.R. McKeown. Beyond semantic ambiguity. In *Proc. of the 7th National Conference on Artificial Intelligence*, pages 751-755, 1988.
- [16] H.T. Ng and R.J. Mooney. On the role of coherence in abductive explanation. In Proc. of the 8th National Conference on Artificial Intelligence, pages 337-342, 1990.
- [17] P. Norvig and R. Wilensky. A critical evaluation of commensurable abduction models for semantic interpretation. In *Proc. of the 13th International Conference on Computational Linguistics*, volume 3, pages 225-230, 1990.
- [18] M. Okumura and H. Tanaka. Towards incremental disambiguation with a generalized discrimination network. In Proc. of the 8th National Conference on Artificial Intelligence, pages 990-995, 1990.
- [19] H. Prüst and M. van der Berg. A coherent approach to underspecification in natural language discourse. In *Proc. of the Workshop on Discourse Coherence*, 1991.
- [20] R. Reiter. A logic for default reasoning. Artificial Intelligence, 13(1-2):81-132, 1980.
- [21] D. Schiffren. Discourse Markers. Cambridge University Press, 1987.
- [22] C.L. Sidner. Focusing in the comprehension of definite anaphora. In M. Brady and R.C. Berwick, editors, Computational Models of Discourse, pages 267-330. MIT Press, 1983.

- [23] M.E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Note 451, SRI International, 1988.
- [24] A. Stolcke. Gapping and frame semantics: A fresh look from a cognitive perspective. In *Proc. of the 13th International Conference on Computational Linguistics*, volume 2, pages 341-346, 1990.
- [25] M. Walker, M. Iida, and S. Cote. Centering in japanese discourse. In Proc. of the 13th International Conference on Computational Linguistics, 1990.
- [26] R.M. Weischedel and N.K. Sondheimer. An improved heuristic for ellipsis processing. In *Proc. of the 20th Annual Meeting of the Association for Computational Linguistics*, pages 85-88, 1982.