Full-text processing: improving a practical NLP system based on surface information within the context

Tetsuya Nasukawa
IBM Research, Tokyo Research Laboratory
1623-14, Shimotsuruma, Yamato-shi, Kanagawa-ken 242, Japan
nasukawa@trl.vnet.ibm.com

Abstract

Rich information for resolving ambiguities in sentence analysis, including various context-dependent problems, can be obtained by analyzing a simple set of parsed trees of each sentence in a text without constructing a precise model of the context through deep semantic analysis. Thus, processing a group of sentences together makes it possible to improve the accuracy of a practical natural language processing (NLP) system such as a machine translation system. In this paper, we describe a simple context model consisting of parsed trees of each sentence in a text, and its effectiveness for handling various problems in NLP such as the resolution of structural ambiguities, pronoun referents, and the focus of focusing subjuncts (e.g. *also* and *only*), as well as for adding supplementary phrases to some elliptical sentences.

1 Introduction

Context processing has been widely recognized as a key technology for improving the accuracy of text analysis, but it has also been considered a high-cost procedure that requires an enormous amount of background knowledge and deep inference mechanisms. It is true that we can always find examples of problems that require common sense and inference mechanisms, such as the classic problems mentioned in (Charniak, 1973), in which the referents of pronouns are not explicitly stated in the text. However, in a text within a restricted domain – particularly in technical documents such as computer manuals – we can observe many context-dependent problems that are solvable without the use of a deep inference mechanism or carefully hand-coded data such as scripts (Schank and Riesbeck, 1981). We therefore tried to develop a practical method that would solve most context-dependent problems and improve the accuracy of text analysis by using a simple mechanism and existing machine-readable data.

To begin with, we developed a framework for processing all sentences in a text simultaneously, so that each sentence can be disambiguated by using information extracted from other sentences within the same text. Without constructing a precise model of the context through deep semantic analysis, our framework refers to a set of parsed trees (results of syntactic analysis) of each sentence in the text as context information. Thus, our context model consists of parsed trees that are obtained by using an existing general syntactic parser. Except for information on the sequence of sentences, our framework does not consider any discourse structure such as the discourse segments, focus space stack, or dominant hierarchy described in (Grosz and Sidner, 1986). Therefore, our approach is fundamentally different from previous approaches to context processing, and is not aimed at obtaining a perfect analysis. However, by extending the unit of the processing object from one sentence to multiple sentences in a source text and by using syntactic information on all the other words in the whole text, such as modifiee-modifier relationships and their positions in the text, our framework improves the overall accuracy of a natural language processing system.

We implemented this framework on an English-to-Japanese machine translation system named Shalt2 (Takeda *et al.*, 1992), and evaluated the framework with various types of technical documents, especially computer manuals. The results have been encouraging. We obtained rich information for resolving ambiguities in sentence analysis, including various context-dependent problems. For example, by assuming a discourse constraint (Gale *et al.*, 1992; Nasukawa, 1993) that polysemous words within a discourse have the same word sense, we can share a result of word sense disambiguation in one sentence with all the words in the discourse that share the same lemma. Furthermore, by assuming a discourse preference, namely, a tendency for each word to modify or be modified by similar words within a discourse, we can obtain clues to determining the modifiees of structurally ambiguous phrases from structural information on all words with the same lemma within the discourse. Moreover, processing a whole text at one time makes it possible to refer to other information such as word frequency and the position of each word, which can be used for resolving pronoun reference and the focus of focusing subjuncts such as *also* and *only*.

In this paper, we describe our robust context-processing method, namely, full-text processing, focusing on its effects on the output of a machine trans-

lation system. In the next section, we briefly describe the framework of our method, which uses a simple context model; then, in the following sections, we illustrate its effectiveness with some actual outputs of our English-to-Japanese machine translation system.

2 Framework

Full-text processing consists of three steps:

1. Generating a context model that consists of parsed trees of each sentence in a source text
2. Refining the context model by assigning a single unified parse tree to each sentence in the text
3. Resolving the problems in each sentence in the context model and generating a final analysis for each sentence in the text

The respective procedures for these steps are described in the following three subsections.

2.1 Generation of a simple context model

In order to refer to context information that consists of data on multiple sentences in a text, it is essential to construct some context model; the first step of the full-text processing method is therefore to construct a context model by analyzing each sentence in an input text. To avoid any errors that may occur during transformation into any other representations, such as a logical representation, we stayed with surface structures, and to preserve the robustness of this framework, we used only a set of parsed trees as a context model. Thus, each sentence of an input text is processed by a syntactic parser in the first step, and the position of each instance of every lemma, its morphological information, and its modifiee-modifier relationships with other content words are extracted from the parser output, and stored to construct a context model, as shown in Figure 1. In addition, if any on-line knowledge resources are available, information extracted from the resources is also stored in the context model. For example, information on synonyms extracted from an on-line thesaurus dictionary and information on word sense and structural disambiguation extracted from an example base, such as one described in (Uramoto, 1991) and (Nagao, 1990), may be added to the context model.

2.2 Refinement of the context model

In the first step, a syntactic parser may not always generate a single unified parse tree for each sentence in the source text. A syntactic parser with general grammar rules is often unable to analyze not only sentences with grammatical errors and ellipses, but also long sentences, owing to their complexity.[1] Thus, it is indispensable to establish a correct analysis for

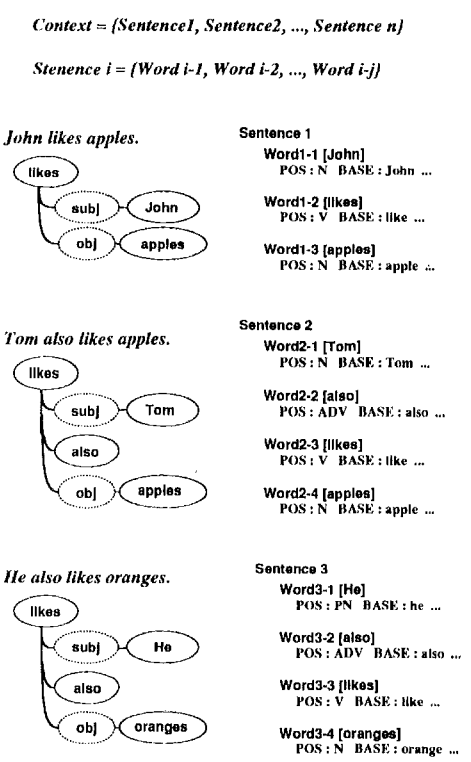


Figure 1: Example of a context model

such a sentence. Information extracted from complete parses of well-formed sentences[2] in a context model can be used to complete incomplete parses, in the form of partially parsed chunks that a bottom-up parser outputs for ill-formed sentences by using a previously described method (Nasukawa, 1995).

On the other hand, for some sentences in a text, such as *Time flies like an arrow*, a syntactic parser may generate more than one parse tree, owing to the presence of words that can be assigned to more than one part of speech, or to the presence of complicated coordinate structures, or for various other reasons. In attempting to select the correct parse of such a sentence, one can use the types of the previous and subsequent sentences or phrases (such as sentence, noun phrase, verb phrase, and so on) and the modifier-modifiee patterns in the context model.

Therefore, in the second step, the context model generated in the first step is refined by referring to information in the context model. First, the most preferable candidate parses are selected for sentences with multiple parses by referring to information on each sentence in the context model for which a parser generated a single unified parse. Then, partial parses of ill-formed sentences are completed by referring to information on well-formed sentences in the context model.

The algorithm for multiple parse selection based on

[1] In texts from a restricted domain, such as computer manuals, most sentences are grammatically correct. However, even a well-established syntactic parser usually fails to generate a unified parsed structure for about 10 to 20 percent of all the sentences in such texts, and the failure in syntactic analysis leads to a failure in the final output of an NLP system.

[2] In this paper, a "well-formed sentence" means one that is parsed as one or more than one unified structure, and an "ill-formed sentence" means one that cannot be parsed as a unified structure.

the context model is as follows:

1. In each candidate parse of a sentence with multiple candidate parses, assign a score for each modifier-modifiee relationship that is found in the context model, and add up the scores to assign a preference value to the candidate parse.
2. Select the parse or parses with the highest preference value. If more than one parse has the highest preference value, go to the next step with those parses; otherwise, leave this procedure.
3. Assign a preference value to each remaining candidate parse that has the same type of root node (such as noun phrase, verb phrase, or sentence) as the parse of the preceding sentence or the next sentence.
4. Select the parse or parses with the highest preference value. If more than one parse has the highest preference value, go to the next step with those parses; otherwise, leave this procedure.
5. Assign a preference value to each remaining candidate parse based on heuristic rules that assign scores to structures according to their grammatical preferability.
6. Select the parse or parses with the highest preference value. If more than one parse has the highest preference value, select the first parse in the list of the remaining candidate parses.

The procedure of completing partial parses of an ill-formed sentence consists of two steps:

1. Inspecting and restructuring of each partial parse
   The part of speech and the modifiee-modifier relationships with other words are inspected for each word in a partial parse. If the part of speech and the modifiee-modifier relationships with other words are different from those in the context model, the partial parse is restructured according to the information in the context model.
2. Joining of partial parses
   If the partial parses were not unified into a single structure in the previous step, they are joined together on the basis of modifier-modifiee relationship patterns in the context model so that a unified parse is obtained.

2.3 Problem resolution for each sentence in the context model

Finally, in the third step, each sentence in the context model is analyzed individually, and its ambiguities and context-dependent problems are resolved by referring to information on other sentences in the context model. The next section describes the procedures for problem resolution, and explains their effectiveness in improving machine translation output.

3 Effectiveness

The accuracy of syntactic analysis may be improved by refinement of the context model in the second step of the procedure. For example, in an experiment on 244 sentences from a chapter of a computer manual, in which we attempted to select the correct parse of a sentence from multiple candidate parses, correct parses were selected for 89.1% of 110 multiple parsed sentences by using information in the context model, whereas the success rate obtained when the context model contained no information was 74.5%. In our experiment on ill-formed sentences in technical documents, in more than half of the incompletely parsed sentences, the partial parses were joined into a single structure by using information in the context model. However, after the second step, ambiguities in each sentence are kept unresolved in the context model. Thus, we need to resolve problems in each sentence in the context model individually.

In this section, we describe how the accuracy of sentence analysis in other problems is improved by referring to the simple context model, and how the results are reflected in improved machine translation outputs.

3.1 Resolving the focus of focusing subjuncts

Resolving the focus of focusing subjuncts such as *also* and *only* is a typical context-dependent problem that requires information on the previous context. Focusing subjuncts draw attention to a part of a sentence that often represents new information. Consider the second sentence, *Tom also likes apples*, in Figures 1 and 2. In this sentence, the scope of *also* can be *Tom*, *likes*, the entire predicate (the whole sentence except the subject *Tom*), or *apples*, according to the previous context. In this case, the preceding sentence, *John likes apples*, has the structure, A *likes* B, whereas sentence (2) has the structure, X *also likes* B, where B and the predicate *likes* are identical. The comparison of these two structures indicates that the new information X (*Tom*) is the scope of *also* in sentence (2).

The focus of focusing subjuncts is resolved by means of the following algorithm:

1. Find among the previous sentences in the context model one that contains expressions morphologically identical with those in the sentence containing the focusing subjunct.
2. Compare each candidate focus word or phrase in the sentence containing the focusing subjunct with words or phrases in the sentence extracted in step 1.
3. Drop any morphologically identical words or phrases as candidates for the focus, and select the remainder as the focus of the focusing subjunct. If more than one candidate remains, take the default interpretation that would be used if there were no context information.

Figure 2 shows the translation outputs of our system with and without information provided by context processing. As shown in this figure, without the context information, *also* modifies the predicate *like* by default in both sentences (2) and (3). In contrast, when context processing is applied, the focus of *also* is determined to be *Tom* in sentence (2) and *orange* in sentence (3).

In our analysis of computer manuals, most nouns were repeated with the same expressions unless they were replaced by pronouns or definite expressions such as *this*, *that*, and *the*. On the other hand, predicates were sometimes repeated with different expressions. For example:

A *has* B. → A *also includes* C.
A *contains* B. → C *is also included in* A.

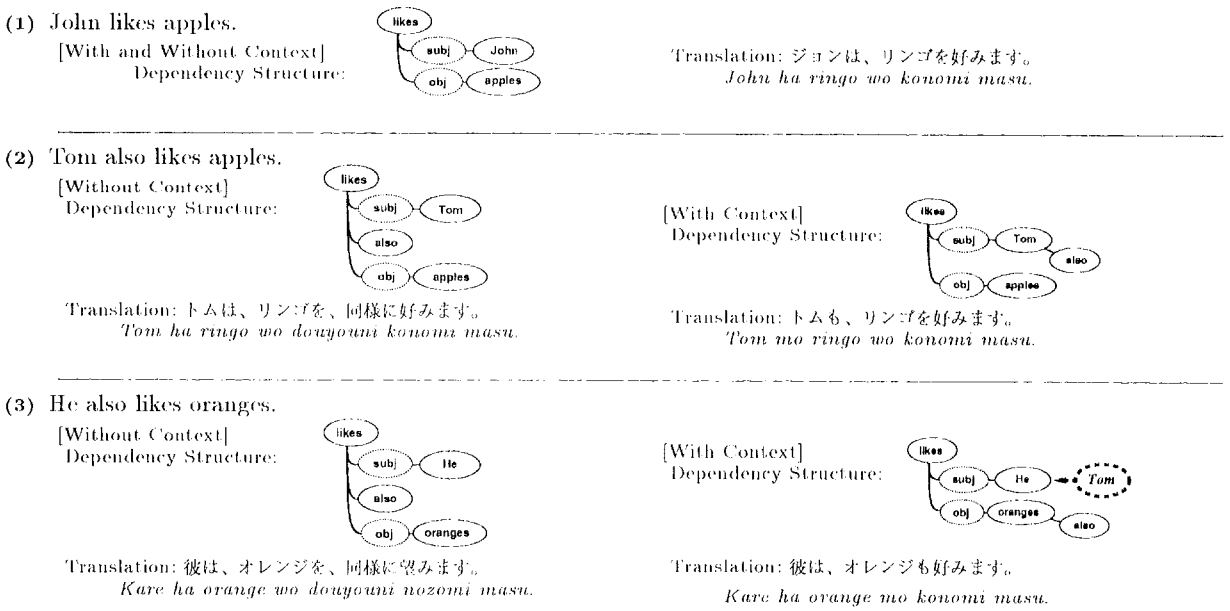


Figure 2: Example of translation (I)

In this case, information on synonyms and derivatives extracted from on-line dictionaries can be used to examine the correspondence between two words.

3.2 Resolving pronoun referents

Pronoun resolution is another typical context-dependent problem, since the referent of a pronoun is not always included in the same sentence. Our context model is used to select candidate noun phrases for a pronoun referent. Furthermore, information on word frequency and modifier-modifiee relationships extracted from the context model improves the accuracy with which the correct referent is selected from the candidate noun phrases, as shown in a previous paper (Nasukawa, 1994). By applying heuristic rules according to which a candidate that has been frequently repeated in the preceding sentences and a candidate that modifies the morphologically identical predicates as the pronoun in the same context are preferred, we obtained a success rate of 93.8% in pronoun resolution.

However, the results of pronoun resolution may not be explicitly reflected in the output of a machine translation system, since most languages have corresponding anaphoric expressions, and use of the corresponding anaphoric expression in the translation output has the advantage of avoiding misinterpretations caused by misresolution of pronoun referents, even if the probability of misinterpretation is less than 10%. Thus, in Figure 2, *He* in sentence (3) is translated as the Japanese pronoun *kare*, although its referent is correctly resolved as *Tom*. Even so, correct resolution of a pronoun referent is important for disambiguating the word sense of a predicate modified by the pronoun.[3] In addition, if the positions of a pronoun and its referent noun phrase are reversed in the translation of a complex sentence where an initial main clause in a source-language sentence comes after the subordinate clause in the target language, the referent noun phrase should be replaced with the pronoun, to avoid cataphoric reference. For example, the English sentence

> *The dog will eat your cake if you don't have it quickly,*

should be translated as

> *Kimi* [You] *ga sono keiki* [the cake] *wo suguni* [quickly] *tabe nai* [don't eat] *nara, sono inu* [the dog] *ga tabete_shimawayo* [will eat].[4]

Since in the translated Japanese sentence the subordinate clause, *if you don't have it quickly*, comes before the main clause, *The dog will eat your cake*, the pronoun *it* in the subordinate clause must be resolved in order to generate a natural Japanese sentence. Moreover, the word sense of *have* in the subordinate clause cannot be selected without information on the referent of the pronoun *it*.

3.3 Lexical and Structural disambiguation

In a consistent text, polysemous words within a discourse tend to have the same word sense (Gale *et al.*, 1992; Nasukawa, 1993). Thus, by applying discourse constraint in such a manner that polysemous words with the same lemma within a context have the same

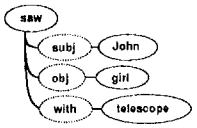
[3] In fact, the result of pronoun resolution for sentence (3) of Figure 2, in which *Tom* is selected as the referent of *He*, is reflected in the translation of the predicate *like*. Because of the lack of a semantic feature *human* for the lexical entries *Tom* and *John* in our dictionary at the time of this translation, different word senses for animate subjects and non-animate subjects were selected for the verb *like*, and the verb *like* was rendered differently in the translations with and without context.

[4] This translation was not produced by our system.

word sense, a result of word sense disambiguation applied in one sentence can be shared with all other words in the context that have the same lemma. Furthermore, by assuming discourse preference, namely, a tendency for each word to modify or be modified by similar words within a discourse, structural information on all other words with the same lemma within the discourse provides clue for determining the modifiees of structurally ambiguous phrases (Nasukawa and Uramoto, 1995). This method can be used to solve context-dependent problems such as the well-known example shown in Figure 3.

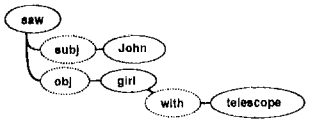
(1) John saw a girl with a telescope.

[Without Context]
Dependency Structure:



Translation: ジョンは、望遠鏡によって、少女を見ました。
John ha bouenkyou niyotte shoujo wo mimashita.

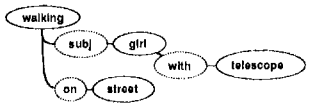
[With Context]
Dependency Structure:



Translation: ジョンは、望遠鏡をもつ少女を見ました。
John ha bouenkyou wo motsu shoujo wo mimashita.

(2) The girl with a telescope was walking on the street.

[With and Without Context]
Dependency Structure:



Translation: 望遠鏡をもつ少女は、通りで歩いていました。
Bouenkyou wo motsu shoujo ha toori de aruite imashita.

Figure 3: Translation with context (II)

In sentence (1) of the figure, the modifiee of the prepositional phrase *with a telescope* can be either *saw* or *girl*, depending on its context. In this case, information in sentence (2), where the identical prepositional phrase modifies *girl*, provides a clue that *with a telescope* in sentence (1) is likely to modify *girl*. In this way, modifier-modifiee relationships extracted from a context model provide clues for disambiguating structurally ambiguous phrases. Needless to say, the effectiveness of this method is highly dependent on the source text, and it may seem too optimistic to expect such useful information in the same context. However, as shown in Figure 4, which is a translation output of an actual computer manual, we can often find modifier-modifiee relationships that disambiguate structurally ambiguous phrases in the same context, at least in technical documents. In Figure 4, the ambiguous prepositional phrase *of a job*[5] in sentence (2) is disambiguated and attached to *the flow* by using the information provided by the unambiguous prepositional phrase in *The flow of a job* in sentence (7). Similarly, the information on the unambiguous prepositional phrase in *placed on an output queue* in sentence (11) disambiguates the ambiguous prepositional phrase *on a job queue* in sentence (9), allowing it to be attached to *places*.

[5] *of* + noun may modify verb, as in *He robbed a lady of her money.*

3.4 Supplementing phrases for elliptical sentences

Supplementation of elliptical phrases is another typical context-dependent problem. In spite of the simplicity of our context model, some elliptical phrases can be supplemented by using information extracted from the context model. For example, if a group of words ending with a colon is not a complete sentence, as in the case of (3) in Figure 4,

This allows you to:

our system adds either *do the following* or *the following* by referring to the type of the next sentence or phrase in the context model. If verb phrases follow, *do the following* is added, and if noun phrases follow, *the following* is added. Thus, in (3) in Figure 4, *do the following* is added because a verb phrase follows this sentence.

3.5 Resolving modality

The modality of itemized sentences or phrases is often ambiguous as a result of the presence of ellipses. For example, (4), (5), and (6) in Figure 4 could be imperative sentences in certain contexts. In this case, however, they are itemized phrases, and by reference to (3), they can be identified as supplementary verb phrases to be attached to (3). Thus our system analyzes them as verb phrases and nominalizes them in the translation.

4 Discussion

We have described how a simple context model that consists merely of a set of parsed trees of each sentence in a text provides rich information for resolving ambiguities in sentence analysis and various context-dependent problems. The greatest advantage of our context-processing method is its robustness. Storing information on a large number of sentences requires a relatively large memory space, which has become available as a result of progress in hardware technology. Our framework is highly practical, since it does not require any knowledge resources that have been specially hand-coded for context processing, or a deep inference mechanism, yet it improves the accuracy of sentence analysis and the quality of a practical NLP system. The basic idea of our method is to improve the accuracy of sentence analysis simply by maintaining consistency in word sense and modifiee-modifier relationship among words with the same lemma within the same text, on the basis of the following assumptions:

- Vocabulary is relatively small in a consistent text, and words with the same lemma are repeated in a relatively small area of a text.

(1) Tracking Your Job

ユーザーのジョブを追跡すること〔*User no job wo tsuisekisuru koto*〕

(2) It is important to know the flow of a job so that you can track it through the system and display or change its status.

ユーザーが、システムを通して、それを追跡できて、およびその状況を表示できるか、あるいは変更可能なように、ジョブの流れを知っていることは重要です。〔*User ga, system wo tooshite, sore wo tsuisekidekite, oyobi sono joukyou wo hyoujidekiruka, aruiha henkou kanouna youni, job no nagare wo shitteiru koto ha juuyou desu.*〕

(3) This allows you to:

これは、ユーザーにとって、以下を行なうことを可能にします。〔*Kore ha, user ni totte, ika wo okonau koto wo kanou ni shimasu.*〕

(4) End or hold a batch job.

バッチ・ジョブを終了することあるいは保持すること〔*Batch job wo shuuryousuru koto aruiha hojisuru koto*〕

(5) Answer messages sent by the system.

システムによって送られるメッセージに答えること〔*System ni yotte okurareru message ni kotaeru koto*〕

(6) Control printer output.

印刷装置の出力を制御すること〔*Insatsusouchi no shutsuryoku wo seigyosuru koto*〕

(7) The flow of a job can have up to five steps:

ジョブの流れに、最大5のステップがあり得ます:〔*Job no nagare ni, saidai 5 no step ga ariemasu:*〕

(8) 1. A user or program submits a job to be run.

1. ユーザーあるいはプログラムは、実行されるためのジョブを実行依頼します。〔1. *User aruiha program ha, jikkousareru tame no job wo jikkouiraishimasu.*〕

(9) 2. The system places the job on a job queue.

2. システムは、ジョブ待ち行列に、ジョブを置きます。〔2. *System ha, jobmachigyouretsu ni, job wo okimasu.*〕

(10) 3. The system takes the job from the job queue and runs it.

3. システムは、ジョブ待ち行列から、ジョブを取り、それを実行します。〔3. *System ha, jobmachigyouretsu kara, job wo tori, sore wo jikkoushimasu.*〕

(11) 4. If this job creates some information (output) that needs to be printed, the printer output is placed on an output queue.

4. このジョブが、印刷される必要があるいくつかの情報(出力)を作成する場合には、印刷装置の出力は、出力待ち行列に配置されます。〔4. *Kono job ga, insatsusareru hitsuyou ga aru ikutsuka no jouhou (shutsuryoku) wo sakuseisuru baai niha, insatsusouchi no shutsuryoku ha, shutsuryokumachigyouretsu ni haichisaremasu.*〕

(12) 5. The system takes printer output from the output queue and sends it to the desired printer to be printed.

5. システムは、出力待ち行列から、印刷装置の出力を取り込み、印刷されるための必要な印刷装置に、それを送ります。〔5. *System ha, shutsuryokumachigyouretsu kara, insatsusouchi no shutsuryoku wo torikomi, insatsusarerutame no hitsuyouna insatsusouchi ni, sore wo okurimasu.*〕

Figure 4: Translation with context (III)

- Polysemous words within a discourse tend to have the same word sense.
- Words with the same lemma tend to modify or be modified by similar words.
- Topical words tend to be repeated frequently.

Therefore, the effectiveness of this method is highly dependent on the source text. However, at least in most technical documents such as computer manuals, the above assumptions hold true, and we have had encouraging results.

Acknowledgements

I would like to thank Michael McDonald for his invaluable help in proofreading this paper. I would also like to thank Taijiro Tsutsumi, Masayuki Morohashi, Koichi Takeda, Hiroshi Maruyama, Hiroshi Nomiyama, Hideo Watanabe, Shiho Ogino, Naohiko Uramoto, and the anonymous reviewers for their comments and suggestions.

References

Eugene Charniak. 1973. Jack and Janet in Search of a Theory of Knowledge. In *Proceedings of IJCAI-73*, pages 337–343.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One Sense per Discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop.*

Barbara J. Grosz and Candance L. Sidner. 1986. Attentions, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204.

Daniel Lyons and Graeme Hirst. A Compositional Semantics for Focusing Subjuncts. In *Proceedings of ACL-90*, pages 54–61, 1990.

Katashi Nagao. 1990. Dependency Analyzer: A Knowledge-Based Approach to Structural Disambiguation. In *Proceedings of COLING-90*, pages 282–287.

Tetsuya Nasukawa. 1993. Discourse Constraint in Computer Manuals. In *Proceedings of TMI-93*, pages 183–194.

Tetsuya Nasukawa. 1994. Robust Method of Pronoun Resolution Using Full-Text Information. In *Proceedings of COLING-94*, pages 1157–1163.

Tetsuya Nasukawa. 1995. Robust Parsing Based on Discourse Information: Completing Partial Parses of Ill-Formed Sentences on the Basis of Discourse Information. In *Proceedings of ACL-95.*

Tetsuya Nasukawa and Naohiko Uramoto. Discourse as a Knowledge Resource for Sentence Disambiguation. In *Proceedings of IJCAI-95*, 1995.

Roger C. Schank and Christopher K. Riesbeck. 1981. *Inside Computer Understanding: Five Programs plus Miniatures.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Koichi Takeda, Naohiko Uramoto, Tetsuya Nasukawa, and Taijiro Tsutsumi. Shalt2: Symmetric Machine Translation System with Conceptual Transfer. In *Proceedings of COLING-92*, pages 1034–1038, 1992.

Naohiko Uramoto. 1992. Lexical and Structural Disambiguation Using an Example-Base. In *Proceedings of the 2nd Japan-Australia Joint Symposium on Natural Language Processing*, pages 150–160.