3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

COOPERATIVE DISTRIBUTED PROCESSING FOR UNDERSTANDING DIALOGUE UTTERANCES

Akira Shimazu Kiyoshi Kogure Mikio Nakano

NTT Basic Research Laboratories
3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-01, Japan
e-mail: {shimazu, kogure, nakano}@atom.ntt.jp

10.21437/ICSLP.1994-28

ABSTRACT

This paper proposes a cooperative distributed natural language understanding model that does syntactic, semantic, and pragmatic analyses independently and in parallel, unifying analysis results (logical forms) with those of other processes and complementing each other. The model is capable of analyzing irregular expressions in spontaneous speech and of processing them in real-time. An experimental system that implements this model demonstrates robust analysis.

1 INTRODUCTION

A dialogue understanding system must handle irregular (ungrammatical and fragmentary) expressions in real-time to be capable of responding naturally to a speaker. Traditional processing models for written text are unsuitable for this task.

Various robust parsing methods for written sentences have been proposed for analyzing ungrammatical sentences, but they are inefficient and treat only written sentences. These methods involve reanalyzing the same sentence one or more times, making real-time processing very difficult. They are thus inappropriate for use in dialogue understanding. Furthermore, since these methods reanalyze a sentence using relaxed constraints, many ambiguities are generally incorporated in the result. Therefore disambiguation is needed to get an appropriate result.

Various efficient parsing techniques have been invented to analyze sentences fast. However, it is not clear how to apply them to the real-time processing of spontaneous speech. Particularly, real-time processing is very difficult if traditional robust parsing techniques are used because of the multiple reanalyses and the ambiguities that result from relaxed constraints.

To cope with these problems, we propose a cooperative distributed natural language processing model, the *Ensemble Model* (EM), in which independent analysis processes run in parallel, unifying each of the analysis results (logical forms) with those of other processes at appropriate points and complementing each other. Each process reflects a different analysis strategy, such as syntactic, semantic or pragmatic strategies, or some combination of them, and produces results from input independently without help of other processes. A process which reflects a syntactic strategy uses syntactic information for analyzing an input sentence and constructing a logical form of the sentence. Similarly, a process which reflects some analysis strategy uses mainly the correspoinding information. Here, we assume that the processings of the various aspects are not mutually exclusive.

Each analysis process of the ensemble model writes its partial results into a shared memory at appropriate points, such as at phrase or sentence boundaries. A partial result means a logical form that corresponds to a part of an input sentence. When partial results are produced within an appropriate time, the model tries to unify the results with corresponding results which have been recorded, and writes the unified results into the shared memory if any are obtained. When only one process succeeds, the results of that process are used. Even if some process fails in analysis or cannot terminate within an appropriate time, the ensemble model uses the results of the other processes and proceeds with the analysis. These characteristics make the ensemble model applicable to robust real-time analysis.

The characteristics of the model are advantages over conventional robust parsing methods. Methods which relax constraints after analysis failure analyze the same sentence more than twice: first with normal grammar and then with relaxed constraints[12, 21, 13]. The methods are effective in finding grammatical errors, but are not appropriate for real-time processing. The method called parse fitting tries to find useful parts in a sentence after parsing failure and combines the partial parses[1, 5, 8]. These methods are weak in determining the correct relations between partial parses. Such a case is treated by the complement process in the ensemble model. The chart-based method [14, 9] is viewed as a combination of the above two methods. Since it is a syntactic method, it is not clear how semantic phenomena are treated. Abduction-based methods have been tried in the analysis of ill-formed sentences [6, 2]. They are a closely interleaved model and are inappropriate for real-time processing. Heuristics have been tried for analyzing Japanese sentences which lack case particles [22]. This approach does not put importance on the semantic aspect and cannot treat semantic consistency.

This paper describes the ensemble model, and explains, as an example of the model, the *Ensemble/Trio-1* system. The *Ensemble/Trio-1* system performs syntactic constraint-centered, semantic constraint-centered, and syntactic-semantic constraint-centered analyses on transcribed spontaneous speech that contains ungrammatical expressions. Before acoustic input of spontaneous speech can be treated, we think the transcribed input must be treated as an intermediate step. To deal with spontaneous language, the *Ensemble/Trio-1* system uses a grammar and a dictionary for spontaneous language[17].

2 COOPERATIVE DISTRIBUTED NATURAL LANGUAGE UNDERSTANDING MODEL

2.1 Relationships among Syntactic, Semantic and Pragmatic Processing

Syntactic, semantic, and pragmatic processings are considered necessary in the understanding of language. For combining these types of processings, a serial analysis (offline) model and an integrated or interleaved analysis (online) model have been proposed. The serial model basically tries syntactic analysis first, applies semantic analysis to that result, and then applies pragmatic analysis to the result of the semantic analysis. In the interleaving model, the analysis processes call each other interactively, as the sentence is processed incrementally. Though there has been much discussion about the respective merits of these models, the question of which is the best remains unanswered.

Considering the necessity of understanding spontaneous utterances and responding to an interlocutor in real time, the offline model is inappropriate, since semantic and pragmatic processing are done after a sentence has been input and, if ungrammatical, input sentences require reanalysis, further adding to the processing time. Nor is the online model too promising, since the total analysis time for an utterance is the sum of the times for syntactic, semantic, and pragmatic processing. Recently, a constraint-based method has been tried[4]. In this method, syntactic, semantic, and pragmatic constraints are represented in a uniform manner. However, this does not change the situation, and this method is also inappropriate.

2.2 Ensemble Model

In EM, independent analysis processes run in parallel, writing partial results to a shared memory, unifying partial results of analysis with each other at appropriate points and complementing each other. Each analysis process reflects various syntactic, semantic or pragmatic strategies or a combination of them, and produces results independently of other processes. A process which reflects a syntactic strategy uses syntactic information for analyzing an input and for constructing a logical form of the input. Similarly, a process which reflects some analysis strategy uses the corresponding information mainly. Here, we do not restrict the processings of the various aspects to mutually exclusive proccessings; they may involve some common processing. The unification operation aims to make results more appropriate. When partial results are produced within an appropriate time, the model tries to unify the results with corresponding results which have been recorded, and adds the results to the shared memory if they can be unified. When only one process succeeds, the results of that process are used. Even if some process fails in analysis or cannot terminate within an appropriate time, the EM uses the results of the other processes and proceeds with the analysis. In some cases, the process which failed in the analysis uses the partial results produced by the successful process and continues the analysis, getting appropriate results. We call this complement processing.

2.3 Language Usage

EM reflects observation of human language usage, in which a particular process mode is mainly used in specific situations, as explained below.

A mother, for instance, understands her baby's condition from his cry, and the baby conveys his needs to his mother with the cry. In other cases, speaker's meaning can also be conveyed and understood by means of even a single word. In the utterances of a baby, a nonnative, a person in a hurry, an intimate friend or a spouse, communication often succeeds with only a few words. In such situations, pragmatic constraints seem to be the primary factor in understanding. In everyday conversation, utterances often lack particles or are syntactically ambiguous. Even such irregular utterances are understood on the basis of semantic relations of words. In such situations, semantic constraints seem to be the main factor. In written sentences, information is sent or received using complex syntax where particles and auxiliary verbs are properly used. Even in spontaneous speech, syntactic expressions such as case particles are important for understanding new information, since the hearer may not have semantic information a priori. In such situations, syntactic constraints seem to be the main factor.

As the above examples show, language understanding processes have several aspects. Emphasis is placed on the processing for the specific aspect that is most appropriate for capturing the characteristics of a particular utterance. For this, we assume that the processing which put emphasis on some aspect, such as the syntactic, semantic or pragmatic aspect, is done independently, and that the analysis results of a specific process mainly serves when other processes are less successful. The assumptions are suitable for real-time processing of irregular expressions.

2.4 Robustness of Ensemble Model

When an input sentence is irregular in that it does not satisfy syntactic conditions of a system's grammar, as when it lacks required particles, traditional analysis methods cannot analyze it. The subsequent processing is then not carried out, and the language processing fails as a whole. Similar situations occur in semantic processing. When there are metaphorical expressions which do not satisfy semantic conditions, semantic analysis fails, and the overall language processing fails even though syntactic analysis succeeds. That is, traditional models treat these conditions conjunctively.

Traditional robust parsing techniques generally regard ungrammatical sentences as exceptions. These techniques first analyze a sentence as if it were grammatical. Then, there are two main methods for handling parsing failures. The first tries to reanalyze the sentence by relaxing constraints such as grammar and dictionary; the second tries to combine partially parsed results, and it can also be regarded as applying relaxed rules to partially parsed constituents. These methods present two problems: relaxed constraints generally bring about ambiguities, and these methods consume much time.

In contrast to traditional analysis methods, the EM can do robust parsing efficiently by virtue of cooperative distributed processing. For syntactically irregular inputs, some aspects of the semantic and pragmatic processing of the model can provide information even if the syntactic aspect of the processings fails. When the semantic aspect cannot be treated because of a lack of knowledge, the syntactic aspect can provide some information.

2.5 Real-timeness of Ensemble Model

The EM is also suitable for real-time processings. As in the processing of ungrammatical inputs, even if some aspects of processing fail in analysis or take excessive time, the other aspects may

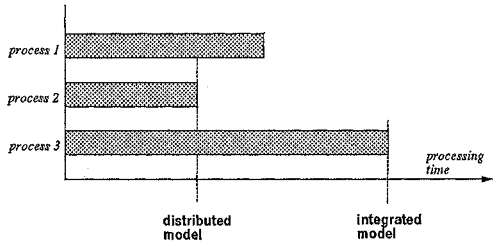

Figure 1: Comparison of a distributed model and an integrated model

produce a result within an appropriate time. Whereas an integrated model does not complete its analysis until all the subprocesses terminate, the EM can produce a result when only some of the processes produce their results. This does not mean that the model stops processing when the first result has been obtained. A process slow in getting results can continue or postpone the analysis. The lower bound of processing time by EM is given by the fastest analysis process, whereas that of an integrated model is given by the slowest analysis process. This is shown in Figure 1.

The observation of person-to-person dialogue also demands real-timeness. According to acoustic data from person-to-person dialogues[18], an interlocutor often utters an interjectory response slightly before the speaker finishes his utterance. The interjectory utterance functions as an indication of receiving, understanding, or agreeing with the speaker's utterance[19]. The observation shows that it is impossible to respond quickly if semantic and pragmatic analyses are not done until the syntactic analysis is finished.

3 COOPERATIVE DISTRIBUTED ANALYSIS SYSTEM FOR SYNTAX AND SEMANTICS

Ensemble/Trio-1 (TRIO) is an experimental system for implementing the EM and studying the basic characteristics of the model (Figure 2). The system currently focuses on syntactic and semantic aspects.

3.1 Architecture

Processes TRIO consists of the following processes: syntactic constraint-based analysis (SynCoA), semantic constraint-based analysis (SemCoA), syntactic-semantic constraint-based analysis (SynSemCoA), input dispatch to analysis processes, and unification of the analysis results.

The input dispatch process sends the characters of an input sentence one by one.

Each analysis process uses its own grammar and dictionary that can treat colloquial expressions, analyzes an input incrementally, and writes partially analyzed results encoded in logical language to a shared memory for appropriate words or phrases. Each type of analysis in TRIO is done using the bottom-up chart method[10], in which the partial sequence of input characters is looked up in the dictionary while lexical edges are constructed. At this stage, misspellings are treated.

Unification and grammatical inference are processed by a logical constraint system[15]. The unification process tries to combine partial results into a consistent and more informative result when they are sent to the shared memory.

Each analysis process has as its subprocess a complement process. This subprocess helps a process that failed in analysis, using information imported from a successful process.

SynCoA uses a grammar and a dictionary without semantic (selectional) constraints to analyze an input and construct a logical form of the input. SemCoA uses a grammar and dictionary which do not have syntactic (surface case-marker) constraints to analyze an input and construct a logical form of the input. SynSemCoA uses a grammar and dictionary which have both syntactic and semantic constraints to analyze an input and construct a logical form of the input. The reason why SynSemCoA is used is that it is more effective than a loose combination of SynCoA and SemCoA when it works properly.

Constraints Our grammars and dictionaries are based on JPSG (Japanese Phrase Structure Grammar)[3]. In the present version, the mother grammar and dictionary are created first and then

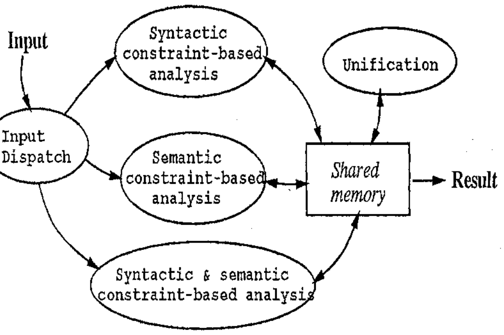

Figure 2: Ensemble system

modified for use in each analysis process. The grammar covers fundamental phenomena in Japanese, such as subcategorization, passivization, causation, topicalization, interrogative, coordination, negation, copula, relative clause, unbounded dependency, and conjunction. The grammar and dictionary for SynCoA are derived from the mother grammar and dictionary by removing sortal constraints; likewise the grammar and dictionary for SemCoA are made from the mother grammar and dictionary by deleting surface case-marker constraints and adding detailed semantic constraints, such as noun-noun semantic relations. Another characteristic of these grammars and dictionaries is that they handle colloquial expressions that appear in spontaneous speech[16].

Output SynCoA, SemCoA, and SynSemCoA processes incrementally and independently analyze an input sentence. Each analysis process appropriately sends its partially derived logical forms (intermediate results) to the shared memory. In the current version, each process sends intermediate results when it recognizes a (surface) speech act[11, 16]. The sending points roughly correspond to the ends of verb or noun phrases. When the analysis has multiple results, they are all sent out to the shared memory.

Unification The unification process compares the intermediate results that cover the same interval of the input sentence, marks them for preference in the shared memory, and records them. Two points must be emphasized regarding unification.

First point is preference. In the present version, all results are recorded with an indication of preference. When SynSemCoA succeeds, its results are preferred over the other because they subsume the results of SynCoA and SemCoA. Unified results are preferred in other cases.

Second point is the unification of results from different sources. Generally, the results of a syntactic constraint-based, a semantic constraint-based, and a pragmatic constraint-based analysis differ in degree of detail. For example, case roles obtained by a syntactic constraint-based analysis are generic, whereas case roles obtained by a semantic constraint-based and pragmatic constraint-based analysis are specific. Therefore, the unification process fills these gaps by inferring case role relations. This is done by the logical constraint system. The gaps in logical forms are filled using axioms expressing case role relations[7].

Complement TRIO has an advantage over traditional robust parsing methods with respect to disambiguation. The traditional system suffers from ambiguities, since relaxed constraints generally bring about many alternative results. TRIO coordinates intermediate results and can thus get appropriate results.

When an analysis process fails in some part of an input but another process succeeds in that part, the process which failed uses the results produced by the successful process and continues the analysis. In the present version, the chart edge created by the successful process is reused and the failed analysis is continued. Here, the edge generally contains ambiguous logical forms, since the successful process has created the edge using relaxed constraints. However, by virtue of the complement processing, the ambiguities can be resolved in the analysis subsequent to the failure, because the analysis uses a different kind of constraints.

3.2 Analysis Examples

Typical analyses produced by TRIO are shown below. Simple sentences are used here for clarity.

Normal Case Each process can analyze the input by itself.

(ex1) Hanako ga Taro o aisuru.
Hanako NOM Taro ACC love
Hanako loves Taro.

SynSemCoA and SynCoA produce the following representation.

love(E1) ∧ agent(E1,X) ∧ hanako(X)
∧ patient(E1,Y) ∧ taro(Y)

Here, E1 is a variable representing a *love* event. agent(E1, X) means X is an agent of E1. The speech act label is omitted from the representation, since it is unnecessary in explaining the example. We will omit the speech act labels in the following examples.

On the other hand, SemCoA does not use syntactic constraints such as surface case-markers, which correspond to case particles in Japanese and prepositions in English, but uses semantic constraints concerning predicates and their semantic roles for "aisuru", "Taro", and "Hanako". This use of constraints brings about the two results that follow.

- love(E1) ∧ agent(E1,X) ∧ hanako(X)
∧ patient(E1,Y) ∧ taro(Y)
- love(E1) ∧ agent(E1,X) ∧ taro(X)
∧ patient(E1,Y) ∧ hanako(Y)

In the first result, "Hanako" is the agent and "Taro" is the patient. In the second, "Taro" is the agent and "Hanako" is the patient. These representations are sent to the shared memory. The unification process tries to unify the logical forms each time they are sent. In this example, the results from SynSemCoA and SynCoA are preferred.

Syntactically Irregular Case SynCoA cannot complete the analysis by itself when the input has omitted words.

(ex2) Hanako ga Taro aisuru.
Hanako NOM Taro love
Hanako loves Taro.

SynCoA and SynSemCoA fail to analyze a sentence which lacks an accusative case particle. Such a phenomenon corresponds to the omission of prepositions in English. In such a case, SynCoA and SynSemCoA fail to combine the noun phrase and the verb. However, SemCoA may succeed using knowledge that represents the meanings of words and their semantic roles. When SemCoA succeeds, SynCoA and SynSemCoA can use those results for the part that they have failed to analyze. Furthermore, SynCoA and SynSemCoA can then continue and resolve the ambiguities that may be in the results from SemCoA.

In this example, SynCoA fails to recognize the verb phrase consisting of "Taro" and "aisuru" (*love*), whereas SemCoA succeeds, making an inactive edge covering "Taro aisuru"(*love Taro* or *Taro loves*). The edge information includes two semantic representations. One represents that "Taro" is the patient, and the other represents that "Taro" is the agent. SynCoA borrows the edge and covers the part which it fails to analyze (Figure 3). In the end, SynCoA and SynSemCoA produce the unique semantic representation in which "Hanako" is the agent and "Taro" is the patient, whereas SemCoA process produces the same two semantic representations as in example 1. As the final result, the system prefers the representation in which "Hanako" is the agent and "Taro" is the patient.

Unlike traditional systems, Trio can sometimes analyze an irregular input faster than it does a normal input. Indeed, it does not necessarily need more time to analyze (ex2) than to analyze (ex1).

Semantically Irregular Case SemCoA process cannot complete the analysis by itself when the constraints of the grammar are not satisfied.

(ex3) Atsugi ga kiso-kenkyu o hajimeru.
Atsugi NOM kiso-kenyu ACC start
Atsugi starts basic research.

This is an example of what is called metonymy. Here, we assume that "Atsugi"(a city near Tokyo) represents "researchers in Atsugi labs". The sentence seems strange out of context, but this kind of utterance often occurs in real dialogues. In this case, SemCoA and SynSemCoA processes fail because the semantic feature of "Atsugi ga" does not match the sortal restriction of an agent role in the subcategorization frame of "hajimeru"(*start*). However, the system treats the sentence as a semantically irregular case. For this sentence, SynCoA process produces the following result.

start(E1) ∧ agent(E1,X) ∧ atsugi(X)
∧ object(E1,Y) ∧ basic-research(Y)

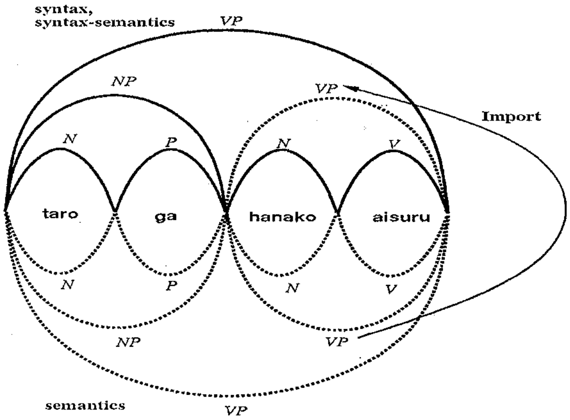

Figure 3: Example of robust analysis
VP(taro aisuru) in the syntax chart is not made from N(taro) and V(aisuru) in the syntax chart, but is imported from the VP edge in the semantics chart.

In this example, this representation is adopted as the final result of the system.

Absorption of Discrepancy between Representations Generally, the results from SynCoA and SemCoA differ in the degree of detail. The unification tries to absorb the discrepancy in such representations by inference[7].

(ex4) Kyoto no eki
Kyoto Gen station
station in Kyoto

For this phrase, the syntactic analysis process outputs the following result.

station(X) ∧ NO-relation(X,Y) ∧ Kyoto(Y)

Here, NO-relation(X,Y) is analyzed from the adnominal particle "no" (roughly *of* in English) [20]. On the other hand, SemCoA and SynSemCoA processes produce the following.

station(X) ∧ location(X,Y) ∧ Kyoto(Y)

The unification process fills the gap between the NO-relation and the location using the logical constraint system, using the following axioms.

building(X) ∧ location(X,Y) ∧ place(Y)
→ NO-relation(X,Y)
station(X) → building(X)
Kyoto(X) → place(X)

Fragmental Expressions and Expressions Peculiar to Spontaneous Speech Expressions peculiar to spontaneous speech are treated with the grammar and dictionary that treat colloquial expressions[16]. At present, we have designed and partially implemented an extended version, *Ensemble/Quartet-1*, to analyze such fragmentary expressions as noun phrases lacking a verb and inverted sentences, by adding a pragmatic analysis process [11].

4 CONCLUSION

This paper proposed the *Ensemble Model* for cooperative distributed natural language processing and explained the experimental system. This model involves parallel, independent analysis processes based on syntactic, semantic, and pragmatic constraints, unification of their analysis results, and complement of each other process. The model appropriately uses required processing aspects, such as syntax, semantics, and pragmatics. This use allows the analysis of irregular expressions in spontaneous speech and real-time processing of them. These are advantages over conventional robust parsing methods.

The ensemble model differs greatly from previous methods for integrated natural language processing systems and robust parsing. Although much work must be done to expand the system into a full-fledged dialogue understanding system, the advantages of the model have been demonstrated by the *Ensemble/Trio-1* experimental system. We are currently in the process of implementing *Ensemble/Quartet-1*, an experimental system that incorporates a plan recognition process for pragmatic constraint-based analysis. The system is implemented in Lucid Common Lisp and the logical constraint system[15]. The parallel processing is simulated using the multitasking facility of Lucid Common Lisp.

REFERENCES

[1] J. G. Carbonell and P. J. Hayes. Robust Parsing Using Multiple Construction-specific Strategies. In L. Bolc, editor, *Natural Language Parsing Systems*, pp. 1–32. Springer-Verlag, 1987.

[2] Y. Den. A Study on a Spoken Dialogue Grammar (in Japanese). *Japanese Society for Artifical Intelligence, SIG-SLUD-9302-5*, pp. 33–40, 1993.

[3] T. Gunji. *Japanese Phrase Structure Grammar.* Reidel, Dordrecht, 1986.

[4] K. Hasida. Sentence Processing as Constraint Transformation. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pp. 339–344, 1990.

[5] P. J. Hayes and G. V. Mouradian. Flexible Parsing. *American Journal of Computational Linguistics*, 7(4):232–242, 1981.

[6] J. R. Hobbs, D. E. Appelt, J. Bear, and M. Tyson. Robust Processing of Real-world Natural-language Texts. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 186–192, 1992.

[7] J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. Martin. Interpretation as Abduction. *Artificial Intelligence*, 63:69–142, 1993.

[8] K. Jensen, G. E. Heidorn, L. A. Miller, and Y. Ravin. Parse Fitting and Prose Fixing: Getting a Hold on Ill-Formedness. *Computational Linguistics*, 9(3–4):147–160, 1983.

[9] T. Kato. Yet Another Chart-based Technique for Parsing Ill-formed Input (in Japanese). In *Information Processing Society of Japan, Natural Language Special Interest Group Technical Report No. 83-10*, 1991.

[10] M. Kay. Algorithm Schemata and Data Structures in Syntactic Processing. Technical report CSL-80-12, Xerox PARC, 1980.

[11] K. Kogure, A. Shimazu, and M. Nakano. Recognizing Plans in More Natural Dialogue Utterances. In *Proceedings of the Third International Conference on Spoken Language Processing*, 1994.

[12] S. C. Kwansy and N. K. Sondheimer. Relaxation Techniques for the Parsing of Unrestricted Texts. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 193–200, 1992.

[13] Y. Matsumoto, Y. Den, and K. Yanagi. A Flexible Natural Language System with Concurrency and Meta-level Processing. In *Proceedings of the Fourth International Workshop on Natural Language Understanding and Logic Programming*, pp. 146–157, 1993.

[14] C. Mellish. Some Chart-based Techniques for Parsing Ill-formed Input. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pp. 102–109, 1989.

[15] M. Nakano. Constraint Projection: An Efficient Treatment of Disjunctive Feature Descriptions. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 307–314, 1991.

[16] M. Nakano, A. Shimazu, and K. Kogure. Parsing Utterances in Dialogue (in Japanese). In *Symposium on Implemented Natural Lanugage Processing.* The Institute of Electronics, Information and Communication Engineers, Japan Society for Software Science and Technology, 1993.

[17] M. Nakano, A. Shimazu, and K. Kogure. A grammar and a parser for spontaeous speech. In *Proceedings of the 15th International Conference on Computational Linguistics*, 1994.

[18] N. Osaka. An Analysis of Address-response Relation in Conversational Speech Centering on Listening Response (in Japanese). SIG-SP87-107, The Institute of Electronics, Information, and Communication Engineers, 1987.

[19] A. Shimazu, M. Kawamori, and K. Kogure. Analysis of Interjectory Responses in Dialogue. In *The Institute of Electronics, Information, Natural Language Understanding and Communication Special Interest Group Technical Report No. 89-15/ Communication Engineers / Information Processing Society of Japan, Natural Language Special Interest Group Technical Report No. 73-9*, pp. 65–72, 1993.

[20] A. Shimazu, S. Naito, and H. Nomura. Semantic Structure Analysis of Japanese Noun Phrases with Adnominal Particles. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pp. 123–130, 1987.

[21] R. M. Weischedel and N. K. Sondheimer. Meta-rules as a Basis for Processing Ill-formed Input. *Computational Linguistics*, 9(3-4):161–177, 1983.

[22] M. Yamamoto, S. Kobayashi, and S. Nakagawa. An Analysis and Parsing Method for the Omission of Post-Positions and Inversion in Japanese Spoken Sentences in Dialog (in Japanese). *Transactions of Information Processing Society of Japan*, 33(11):1322–1330, 1992.