Parsing Japanese Spoken Sentences Based on HPSG

Kiyoshi KOGURE

ATR Interpreting Telephony Research Laboratories
Sanpeidani, Inuidani, Seika-Cho, Soraku-gun, Kyoto 619-02, Japan
kogure%atr-la.atr.junet@uunet.uu.net

ABSTRACT

An analysis method for Japanese spoken sentences based on HPSG has been developed. Any analysis module for the interpreting telephony task requires the following capabilities: (i) the module must be able to treat spoken-style sentences; and, (ii) the module must be able to take, as its input, lattice-like structures which include both correct and incorrect constituent candidates of a speech recognition module. To satisfy these requirements, an analysis method has been developed, which consists of a grammar designed for treating spoken-style Japanese sentences and a parser designed for taking as its input speech recognition output lattices. The analysis module based on this method is used as part of the NADINE(Natural Dialogue Interpretation Expert) system and the SL-TRANS (Spoken Language Translation) system.

1. INTRODUCTION

An analysis module for a spoken sentence translation system, or an interpreting telephony system requires the following capabilities:

(i) the module must be able to treat spoken-style sentences; and,

(ii) the module must be able to accept not only strings but also lattice-like structures where the analysis module directly drives a speech recognition module (e.g., a phoneme or word recognition module but not a whole sentence recognition module) or where the analysis module takes as its inputs partial speech recognition results including both correct and incorrect sentence constituents.

To satisfy these requirements, an analysis method has been developed which consists of a grammar framework designed for treating spoken-style Japanese sentences and a unification-based parser designed for taking as its input speech recognition result lattices.

The grammar framework is unification-based lexico-syntactic and is essentially based on HPSG[10] and JPSG[2]. This is because:

(i) a lexico-syntactic approach is modular in the sense that most of the grammatical information is to be specified in descriptions of lexical items; and that it is therefore easy to extend a grammar simply by adding new lexical items to the lexicon or adding new information to lexical items; and

(ii) the JPSG framework can essentially capture constraints between complex predicate constituents and their complements. This capability is important because spoken-style Japanese sentences often have complex predicate constituents.

The grammar framework is extended from these grammatical frameworks by introducing features related to semantic and pragmatic constraints[12].

The parser developed is essentially based on the active chart parsing algorithm[11] because the algorithm is as efficient as Earley's algorithm[1] or any other CFG parsing algorithm and, moreover, has the capability of controlling parsing strategies to avoid exhaustive searches. The parser is extended to treat constraints in Typed Feature Structures (TFS) by using TFSP links (as defined in Section 3).

The analysis method proposed in this paper is used in the analysis module of the NADINE system[4,9] and the NADINE system is used as the machine translation module of the SL-TRANS system. In the SL-TRANS system, input speech is recognized by the Japanese *bunsetsu*[1] phrase recognition module based on the HMM-LR method[8] and the module outputs the sequence of *bunsetsu* phrase lattices, each of which consists of bunsetsu phrase structure candidates. The outputs are filtered by a *bunsetsu* dependency filter module[5] which outputs sentence lattices consisting of fewer *bunsetsu* phrase structure candidates than the HMM-LR produces.

The NADINE system takes as its input a sentence lattice and outputs an English sentence. The analysis module based on this paper's method takes a sentence lattice and outputs typed feature structures which represent syntactic, semantic and pragmatic information of the sentence. Then, the transfer and generation modules output an English sentence.

In this paper, Section 2 describes the grammar framework and Section 3 describes the parser and the analysis method.

2. GRAMMAR FRAMEWORK FOR SPOKEN-STYLE JAPANESE SENTENCES

The grammar built up to analyze spoken-style Japanese sentences is essentially based on HPSG and JPSG. The grammar describes not only syntactic and semantic information but also discourse and pragmatic information in an integrated way by using TFS descriptions.

Resolution of omitted obligatory cases (or zero-pronouns) is very important because

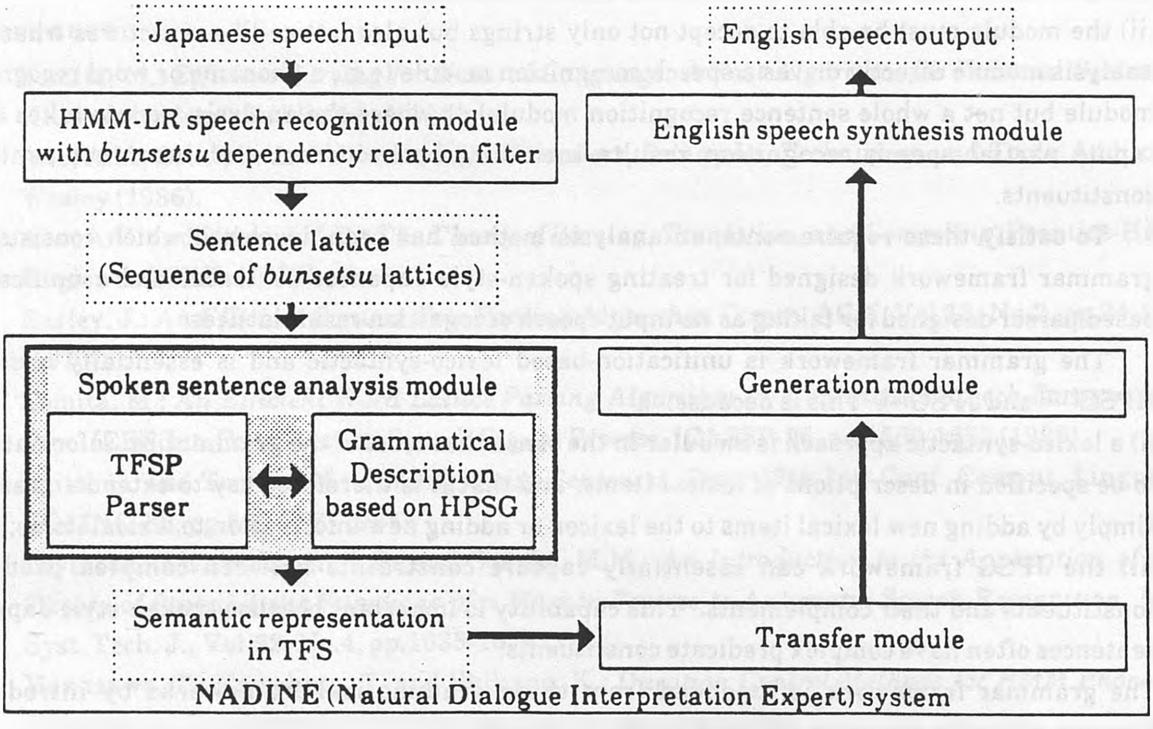
Japanese speech input
HMM-LR speech recognition module with *bunsetsu* dependency relation filter
Sentence lattice (Sequence of *bunsetsu* lattices)
Spoken sentence analysis module
TFSP Parser
Grammatical Description based on HPSG
Semantic representation in TFS
Transfer module
Generation module
English speech synthesis module
English speech output
NADINE (Natural Dialogue Interpretation Expert) system

Fig.1 Overview of the SL-TRANS system (modules related to the analysis module)

[1] a basic phonological phrase consisting of a *jiritsugo*-word such as a noun, verb, or adverb followed by zero or more *fuzokugo*-words such as auxiliary verbs, postpositional particles, or sentence final particles.

(i) pronouns referring to the speaker and the hearer seldom appear in spoken-style sentences and these omitted cases make sentences more ambiguous, and

(ii) in order to translate these sentences into natural English sentences, they must be supplemented.

If they are not supplemented, for example, Japanese sentences without agent subject case expressions must often be translated into unnatural English passive sentences (e.g., "*A registration form will be sent*" instead of "*I will send you a registration form*"). In this paper's analysis, such omitted cases are resolved by using constraints on the uses of deictic expressions and their case elements, and so on.

2.1. Treatment of Syntactic and Semantic Information

Spoken-style Japanese sentences often have complex sentence final predicate phrases consisting of main predicates and combinations of auxiliary verbs and sentence final particles. In such a predicate phrase, its head constituent stipulates the properties of the complement occurring just on its left such as its part of speech, conjugational type, and conjugational form. Such stipulations are easily described in the SUBCAT feature value in the head. A SUBCAT feature value is a list of complement constituent specifications.

For example, in the lexical description (1) of the causative auxiliary verb "*seru*", the SUBCAT feature value specifies that the auxiliary takes as its complement a verb phrase with conjugational type CONS (for consonant type) and conjugational form VONG (for voice negative type), and two postpositional phrases (PPs), a PP marked by "*ni*" and a PP marked by "*ga*". Moreover, it specifies that the VP must be located just before the auxiliary and that the relative order between two PPs is free. The SEMF feature, which is a bundle of semantic features, specifies the semantic selectional restrictions and, in the description, the SEMF feature value of the *ga*-PP specifies that the PP must refer to an animate object.

```
[[syn [[morph [[ctype vow][cform aspl-or-infn]]]
       [head [[pos v]
              [modl [[caus +]]]
              ...]]
       [subcat [[first [[syn [[morph [[ctype cons][cform vong]]]
                              [head [[pos v]
                                     [modl [[caus -][deac -] ...]]]]
                              [subcat [[first [[syn [[head [[form ga]
                                                      ...]] ...] ...]
                                               [sem ?causee]]]
                                       [rest end]]]]]
                        [sem ?caused]]]
                [rest (:perm-list [[syn [[head [[form ga] ...]] ...]]
                                    [semf [[human +]]]
                                    [sem ?causer]]
                                   [[syn [[head [[form ni] ...]] ...]]
                                    [sem ?causee]])]]] ...] ...]
 [sem [[relation cause]
       [causer   ?causer]
       [causee   ?causee]
       [caused   ?caused]]]]                                          (1)
```

where "?" is the prefix of the tag and structures denoted by the same tag are token identical, and ":perm-list" is a macro which takes as its arguments a set of typed feature structure descriptions and returns as its value the disjunction of permuted lists made of the set.

Furthermore, the COH feature (Category Of Head) in a complement or adjunct constituent specifies its head constituents. Combinations of COH and SUBCAT features allow flexible grammatical descriptions.

Japanese predicate constituents belong to groups: a member of these groups must, with some exceptions, occur in a strictly one-dimensional sequence; these groups correspond to semantic hierarchies. A new head feature MODL (for modality) has been devised to all and only predicates with grammatically ordered constituents. For example, in the above description (1), the MODL feature value of the first SUBCAT value element specifies that the complement verb phrase should not include any auxiliary verbs.

Besides the predicate constituent order specification, the MODL feature is also used to restrict syntactic and semantic behavior of subordinate (adverbial) phrases. For example, certain formal adverbs (i.e., subordinate conjunctions) require as their complements verb phrases without time or place modifiers. Such requirements reduce ambiguities of adverbial phrase modificands. The MODL feature in conjunction with the SEMF feature contribute to reducing the number of verbal modificand ambiguities.

2.2. Treatment of Pragmatic Constraints on Uses of Expressions

This grammar framework treats discourse or pragmatic constraints on uses of expressions in order to select plausible analysis candidates and to resolve certain kinds of zero-pronouns. An analysis candidate includes not only syntactico-semantic descriptions such as a semantic interpretation (the SEM feature value) but also annotations or a set of conditions under which the interpretation is valid. For example, the sentence

Watashi	ni	tourokuyoushi	o	o-okuri	itadake	masu	ka
I	DAT	registration form	ACC	HON-send	RECEIVE-FAVOR	POLITE	QUESTION

seems to have two analysis candidates corresponding to phrase structures (a) and (b) in Fig.2 (they correspond to "*Could you please send me a registration form?*" and ??"*Could I please send a registration form?*"). However, the analysis candidate corresponding to (b) has the following annotations:

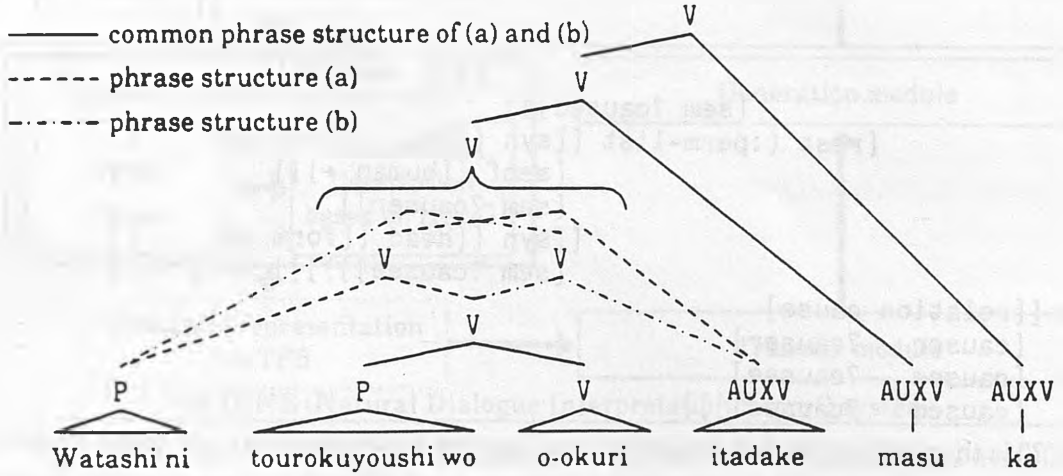

Fig.2 Two derivation trees of the sentence
"watashi ni tourokuyoushi o o-okuri itadake masu ka"

```
[[relation           condescend]
 [agent              ?speaker]
 [object             ?subject_sem]
 [comparative-object ?speaker]]
[[relation           express-more-empathy]
 [agent              ?speaker]
 [object             ?subject_sem]
 [comparative-object ?speaker]]
```

(where ?speaker refers to the speaker and ?subject_sem is the semantic representation of the subject of "*itadake*").

Accordingly, these conditions are unnatural (e.g., the speaker expresses more empathy to a person other than himself) but (a) does not have such unnatural conditions. Thus, the analysis (a) is selected as a more plausible candidate than (b).

These annotations are also used for zero-pronoun resolution. In the analysis (a), the subject and indirect object of "*itadake*" are missing. However, (a) has the following annotations:

```
[[relation           condescend]
 [agent              ?speaker]
 [object             ?subject_sem]
 [comparative-object ?indirect-object_sem]]
[[relation           express-more-empathy]
 [agent              ?speaker]
 [object             ?subject_sem]
 [comparative-object ?indirect-object_sem]]
```

and by searching for discourse participants satisfying these conditions, candidates of missing elements can be found.

In order to obtain such annotations, lexical descriptions have PRAG| RESTRS features which include constraints in terms of **RESPECT**, **CONDESCEND**, **POLITE**, **EXPRESS-MORE-EMPATHY** and so on.

Plausibility scores based on these annotations are used in conjunction with other kinds of scores described below to select plausible analysis candidates. Zero-pronoun resolution is applied after parsing Annotations are used in conjunction with conditions under which utterances of sentences are interpreted as certain types of illocutionary acts, and conditions under which actions in general are rational.

3. FEATURE STRUCTURE PROPAGATION PARSER

3.1. Active Chart Parser with Feature Structure Propagation Links

The active chart parsing algorithm has properties suitable for parsing natural language efficiently. In particular, it has two excellent properties for treating speech recognition result lattices:

(i) it does not limit its inputs to only strings but can accept lattice structures — thus, it can parse speech recognition result lattices directly; and,

(ii) it has the capability of controlling the order of parsing by adapting a method of selecting pending edges from the pending edge list, which works as an agenda. Thus, by adapting a selection method based on certain criteria which, at least, reflects speech recognition result plausibility, plausible parses can be obtained in the early stages without exhaustive search.

However, this second property makes structure sharing difficult in unification-based CFG parsing, or CFG parsing augmented by constraints described in typed feature structures (TFSs). In unification-based parsing, there often exist edges with the same content except for their TFSs. When an active edge is continued with an inactive edge, if there is already an edge with the same contents except for its TFSs as the continuation edge, edge sharing may seem to be able to be achieved by adding the continuation edge's TFSs into the existing edge's. However, this makes parsing incomplete because the existing edge may have been used previously to construct larger edges due to the parsing order freeness and because newly added TFSs are not used to construct larger edges or used as part of larger edges.

In order to solve this problem, the **TFS Propagation parser** (in short, **TFSP parser**) has been developed. The parser is essentially based on active chart parsing and each edge of the parser has a set of TFSs representing syntactic, semantic and pragmatic information of corresponding partial phrase structures. The parser is extended to have special links called **TFS** Propagation links (TFSP links).

A TFSP link in an edge remembers how the TFSs of the edge were previously propagated and specifies how TFSs newly added into the edge should be used. That is, a TFSP link of an active edge points to a continuation edge having as its annotation the inactive edge used to construct the continuation edge. Then, when a TFS is added to an active edge, for each TFSP link of the edge, the TFS is unified with each TFS of the link's inactive edge and then the unification result TFS is added into the link's continuation edge if the unification succeeds. By using TFSP links, new edge creation is necessary only when there is no edge with a certain starting vertex, ending vertex, label and remainder symbol sequence. The TFSP link makes edge structure sharing possible.

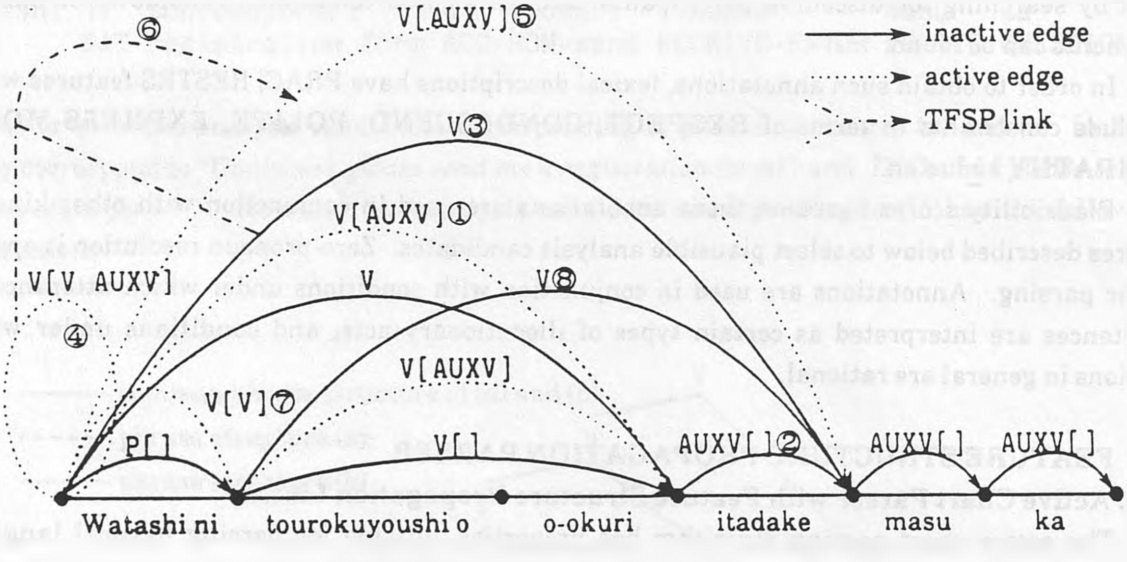

Fig.3 TFSP links

Suppose the case where the inactive edge ③ has been created from the active edge ① and the inactive edge ② and the inactive edge ⑤ has been created from the active edge ④ and the inactive edge ③. The TFSP link ⑥ is created between ④, ③ and ⑤. In this case, when the active edge ⑦ is continued with the inactive edge ⑧, the successful unification result TFSs of ⑦'s and ⑧'s TFSs are added to the edge ③. The edge has a TFSP link and then the newly added TFSs are unified with TFSs in ④ and the successful unification results are propagated to the edge ⑤ as specified by the TFS link ⑥. If there are already TFS links in the edge ⑤, the newly added TFSs are also propagated in the ways specified by these links.

The TFSP link enables the parser to reduce unnecessary edge structure creation and TFS unification. When an active edge is continued with an inactive edge, the continuation edge is meaningful only when it has at least one consistent TFS corresponding to the continuation edge. Therefore, the necessary computation is reduced to finding a pair of active and inactive edge TFSs which are consistent or can be unified. It is not necessary to compute the other pairs' unification after finding a first pair unless TFSs representing whole sentence structures are required later. This is made possible by using TFSP links because they can not only unify TFSs immediately and propagate unification result if desired, but they can also propagate information on how to unify them later. This reduces unnecessary unification computation when the edges are not used as parts of the parses of the whole sentences, especially when the TFSP parser does not need to find all possible parses exhaustively.

The unification method used in the TFSP parser has the following characteristics:
(1) It uses Kasper's disjunctive feature structure unification algorithm[6]. This allows not only for efficient descriptions of each lexical item (such as efficient coding of SUBCAT feature values for treating complement order scrambling and word meanings with conditions for disambiguation), but also packing descriptions of homonyms. Disjunctive lexical descriptions work like Polaroid words[3].
(2) As for the definite feature structure unification algorithm, the incremental copy unification algorithm which allows cyclic structures[7] is adopted to treat cyclic constraints including SUBCAT and COH features.

3.2. Agenda Control Mechanism and Plausibility Score

In order to select the most plausible analysis candidate in the early stages, the TFSP parser selects the pending edge with the best edge score among the pending edge list during parsing, and selects the TFS with the best TFS score among sets of TFSs in complete edges, each of which has as its label the start symbol, as its remainder symbol sequence an empty sequence, as its starting vertex the leftmost vertex of the chart, and as its ending vertex the rightmost vertex of the chart just after parsing finishes. Parsing finishes when a certain number of TFSs have been created with scores better than certain criteria determined by the input sentence length (e.g., the number of *bunsetsu* structures).

The edge score mainly contributes to first obtaining a plausible syntactic structure. The edge score for treating speech recognition result lattices is essentially based on the following:
(a) speech recognition score,
(b) surface string length, and
(c) edge type such as active, inactive, or just-proposed.
When a new edge is created, the edge score is calculated from information on the active edge and the inactive edge. Moreover, when a new TFSP link is created and the links point to an existing continuation edge, the edge score of the continuation is recalculated.

The TFS score mainly contributes to obtaining syntactico-semantically and pragmatically plausible structure and is essentially based on the following:
(d) phrase structure complexity (the number of phrase structure tree nodes),
(e) unfilled complements (the number of elements in SLASH feature value), and
(f) violation of pragmatic constraints on expression usage (the unnatural relationships in the PRAG|RESTRS feature value).

The behavior of the TFSP parser is illustrated by an example. Suppose the case where a speech recognition result lattice includes the following sentence candidates and the nominative postposition "*ga*" has a better speech recognition score than the topic marker "*wa*'" (Fig.4). The parser first tries to build up the structure including "*ga*" due to the speech recognition score preference because there are no other differences between structures including "*ga*" and "*wa*". However, the building-up process stops when combining structures corresponding to "*tourokuyoushi ga*" and "*o-okuri*" because of TFS unification failure between SEMF feature values of the verb's subject [[animate +]] and the nominative noun phrase [[animate -]]. Then, the parser adopts the structure containing "*wa*" and analyzes the semantics of the topic noun phrase as playing a semantic object role in the "*okuru*" (sending) relationship.

In this case, the agent subject is missing and the parser outputs as the semantic representation:

```
[[relation  okuru-1]
 [agent     ?subject_sem]
 [recipient ?indirect-object_sem]
 [object    [[parameter ?x]
             [restriction [[relation tourokuyoushi-1]
                           [object   ?x]]]]]]
```

However, the parser also outputs pragmatic constraints on the person referred to by the subject based on the lexical descriptions of the honorific verb "*itashi*" as follows:

```
[[relation           condescend]
 [agent              ?speaker]
 [object             ?subject_sem]
 [comparative-object ?indirect-object_sem]]
```

After parsing, the analysis module searches for the person to whom the speaker can condescend, and if there is no person other than the speaker and the hearer in the discourse of the utterance, the missing subject is analyzed as referring to the speaker. Then, the following semantic representation is obtained:

```
[[relation  okuru-1]
 [agent     ?speaker]
 [recipient ?hearer]
 [object    [[parameter ?x]
             [restriction [[relation tourokuyoushi-1]
                           [object   ?x]]]]]]
```

From this semantic representation, the output sentence "I send you a registration form." is obtained.

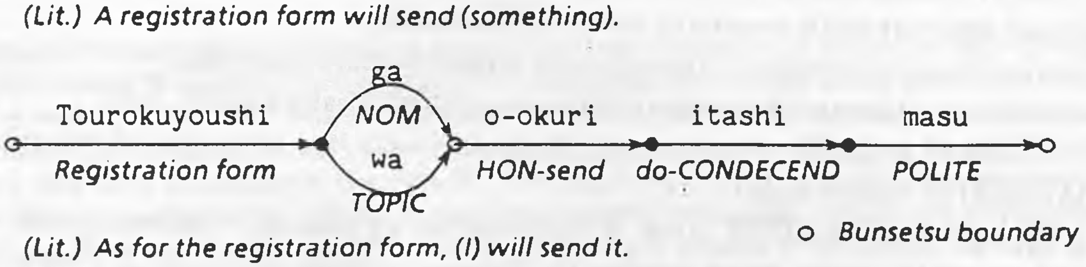

Fig.4 Example of speech recognition result lattice sequence (simplified).

3.3. Experiments

This analysis method is applied to speech recognition results of sentences in 2 task-oriented dialogues about "the secretarial service of the international conference". The HMM-LR speech recognition module with a *bunsetsu* dependency filter outputs for each spoken sentence a sequence of *bunsetsu* phrase lattices. These 2 dialogues consist of 37 sentences. The speech recognition module outputs correct results (i.e., sequences of *bunsetsu* lattices each of which includes the correct *bunsetsu* structure) for 35 sentences. This analysis method is applied to these 35 sentences.

These sentences consists of 76 *bunsetsu* phrases and 112 *bunsetsu* structure candidates. That is, a *bunsetsu* phrase has about 1.47 *bunsetsu* structure candidates.

For this experiment, a grammar was prepared which includes not only lexical items required for accepting correct *bunsetsu* structures in the dialogue, but also all lexical items consisting of all *bunsetsu* structure candidates. The grammar consists of 13 general rules including morphological rules and about 300 lexical entries.

The analysis method obtains correct sentence analysis results for 34 sentences; adequate English sentences are obtained from these correct analysis results. The sentence recognition rate of this method is about 97% and the total sentence recognition rate including the HMM-LR speech recognition module is 92%. The single incorrect analysis result structure, which corresponds to the Japanese sentence "*tourokuyoushi mo o-okuri itashi masu*" (lit. "*I will send you a registration form, too*") instead of "*tourokuyoushi o o-okuri itashi masu*" (lit. "*I will send you a registration form*"), includes as the incorrect speech recognition part only an incorrect modal particle "*mo*" with a higher speech recognition score than the correct case particle "*o*", and the incorrectly recognized structure is perfectly grammatical. In this case, to obtain the correct result requires taking account of the differences in presuppositions derived from these particles and comparing these presuppositions with the context of the utterances.

4. CONCLUSION

In this paper, a new analysis method is proposed for Japanese spoken sentences using a grammar framework for treating spoken-style Japanese sentences and a new parser called the TFSP parser. The grammar framework is essentially based on HPSG and JPSG, and is designed to treat not only syntactic and semantic information but also pragmatic information. Analysis results based on this framework include semantic interpretations of input sentences with annotations on constraints on the uses of these sentences. The TFSP parser has been developed to allow edge structure sharing in unification-based analyses. This method is used as the analysis module of the NADINE system and the SL-TRANS system.

The analysis method is applied to HMM-LR speech recognition result lattices. In parsing lattices, selecting the pending edge with the best score allows the parser to first find plausible candidates. Constraints described in TFSs filter out syntactically or semantically ill-formed structures. The experimental results show that this method is effective in sentence speech recognition. In the experiments, recovering from incorrect recognition requires utterance context understanding including understanding of utterance presuppositions.

ACKNOWLEDGEMENTS

The author is deeply grateful to Dr. Kurematsu, the president of ATR Interpreting Telephony Research Laboratories, Teruaki Aizawa, the head of the Natural Language Understanding

Department, Kei Yoshimoto, and the members of the Natural Language Understanding Department for their constant help and encouragement.

REFERENCE

[1] Earley J.: *An Efficient Context-Free Parsing Algorithm*, Comm. ACM, Vol. 13, No. 2, 1970.

[2] Gunji, T.: *Japanese Phrase Structure Grammar - A Unification-Based Approach*, Dordrecht, Reidel, 1987.

[3] Hirst, G.: *Semantic interpretation and the resolution of ambiguity*, Cambridge University Press, 1987.

[4] Iida, H. et al.: *An Experimental Spoken Natural Dialogue Translation System Using a Lexicon-Driven Grammar*, Proceedings of the European Conference on Speech Communication and Technology, 1989 (to be published).

[5] Kakigahara, K. and Morimoto, T.: *A Study of Bunsetsu Selection Based on the Kakariuke-Dependency*, (in Japanese), IPSG Spring Meeting, 1989.

[6] Kasper, R.: *A Unification Method for Disjunctive Feature Descriptions*, Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics, 1987.

[7] Kato, S. and Kogure, K.: *Efficiency of Feature Structure Unification Methods*, (in Japanese) Proceedings of the Natural Language Working Group of IPSJ, NL64-9, 1987.

[8] Kita, K. et al.: *Parsing Continuous Speech by HMM-LR Method*, Proceedings of the international Workshop on Parsing Technologies, 1989.

[9] Kogure, K. et al.: *A Method of Analyzing Japanese Speech Act Types*, Proceedings of the 2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, 1988.

[10] Pollard, C. and Sag, I.: *Information-Based Syntax and Semantics - Volume 1 Fundamentals*, CSLI Lecture Notes, No. 13, 1988.

[11] Winograd, T.: *Language as a cognitive process - Volume 1 Syntax*, Addison-Wesley, 1983.

[12] Yoshimoto, K. and Kogure, K.: *Phrase Structure Grammar for Inter-Terminal Dialog Analysis*, (in Japanese), IPSG Fall Meeting, 1988.