Incremental Japanese Spoken Language Generation in Simultaneous Machine Interpretation

Koichiro Ryu[†], **Atsushi Mizuno**[††], **Shigeki Matsubara**[†††],
and **Yasuyoshi Inagaki**[††††]
[†]Graduate School of Engineering, Nagoya University
[††]Graduate School of Information Science, Nagoya University
[†††]Information Technology Center/CIAIR, Nagoya University
[††††]Faculty of Information Science and Technology, Aichi Prefectural University
Furo-cho, Chikusa-ku, Nagoya, 464-8601
E-mail:k_ryu@inagaki.nuie.nagoya-u.ac.jp

Abstract

Towards smooth and natural multilingual speech communication, it is desired to develop simultaneous interpretation systems. This paper proposes a technique for incremental Japanese generation in English- Japanese simultaneous machine interpretation. Generally, the difference of word-order in both languages causes a fatal problem in incremental translation between English and Japanese. In our approach, the acceptable translation result can be generated as immediately as possible by positively utilizing inversions, repetitions, corrections, etc. The simultaneous interpretaion can be attained by synchronous execution of incremental dependency parsing, transfer and generation. The prototype system of Enligh-Japanese incremental translation was developed based on our technique. An experiment using ATIS corpus has shown our technique to be effective for improving the output-timing of translation results.

1 Introduction

In order to provide an environment supporting natural and smooth dialogues between different languages, it has been expected to develop simultaneous interpreting systems. Actually, there are several studies on simultaneous interpreting system (J. W. Amtrup, 1999; S. Matsubara, 1997; H. Mima, 1998). We have already proposed a simultaneous interpreting system based on transfer of syntactic structure and have developed its prototype system of English-to-Japanese interpreting *LINAS LINAS* parses the dependency structure of English strings generated by speech recognition. According to it, the *LINAS* determines the word order of Japanese translation and excutes lexical conversion.

LINAS generates translations fulfilled Japanese dependency constraints and based on the basis of the word order of input English utterances. Therefore, *LINAS* generates an acceptable translation result. However, on the other hand, due to the powerful acting of the constraints of the output language, there is a delay between the input and output timing. Namely, our conventional technique of Japanese translation generation in the *LINAS* has adopted dependency grammar as the grammar of a translation. As a result, it is difficult for the system to output a translation simultaneously.

In this paper, we propose a Japanese translation generation technique utilizing inversions. The inversion is to permit that a dependency is directed from right to left. It is possible to generate a translation with more high simultaneity by permitting a translation including inversions. Moreover we examine tendency of inversions in speech dialogue corpus and acquire the feature of inversion from it. Fulfilling the feature makes it possible to generate the acceptable translation result. Our technique using the inversion is introduced into the *LINAS*. An experiment using ATIS corpus (M. P. Marcus, 1993) has shown our technique to be effective for improving the output-timing of translation results.

This paper is organized as follows: Section 2 explains an approach of incremental Japanese generation utilizing inversions and corrections. Section 3 describes the overview of simultaneous interpreting system *LINAS*. Section 4 describes our method of a generation utilizing inversions. And an experiment using our method is presented in Section 5.

2 Incremental Japanese Generation

In this section, We explain our method that the acceptable translation result can be generated as immediately as possible by positively utilizing inversions,

2.1 Incremental Japanese Generation Utilizing Inversions

Let us consider the following English:

(1) I'll go to the park with your brother

The standard Japanese for (1) is

(2) anata-no ani-to (with your brother) koen-ni (to the park) iki-masu (I'll go)

In this case, it is not until the input of "with your brother" that translation can be started. However, if Japanese

(3) koen-ni (to the park) iki-masu (I'll go), anata-no ani-to (with your brother)

is generated as its translation, the output of "koenni iki-masu" can be started as soon as the input of "to the park" is finished. The Japanese sentence (3) includes an inversion phenomenon. Although no dependency is directed from right to left in Japanese dependency grammar, the deviation from the principle occurs frequently in the case of spoken language, and generally such a phenomenon is called inversion. In our research, the inversion is positively used for translation, and therefore less delay can be achieved in simultaneous interpretation.

2.2 Features of Inversion in Spoken Language

However, the inversion cannot necessarily be generated in any dependency, and there are some tendencies in the appearance with the characteristic of bunsetsus. We investigated the appearance tendency of inversion using a Japanese spoken language corpus with dependency structures which was constructed at CIAIR, Nagoya University (T. Ohno, 2003). In addition, by this research, the dependency from right to left was defined as inversion. The following character became clear as a result of observing all the 230 inversion phenomena included in 7781 utterances (11789 bunsetsu).

- The type of head bunsetsus is predicate.
- The number of bunsetsus which depend a head bunsetsu is two or more corresponds to a basic phrase in English.
- The part-of-speech of the last morpheme of dependent bunsetsus is particle, noun, adverb, etc. The kind of particle is "-wa", "-de", "-ni", etc.
- The number of inversion in a utterance is at most one.

2.3 Japanese Generation Utilizing Corrections

In our method, the inversion can be used for Japanese generation only when the above-mentioned character is fulfilled. However, generally it is not decided to the last of an input sentence whether the character is fulfilled or not. For this reason, when it becomes clear not to fill above-mentioned restrictions after generating inversion, the generation of inversion is avoided by generating a bunsetsu again. For example, when the complete English of (1) is

(4) I'll go to the park with your brother by his car next Sunday.

in fact, it generates

(5) koen-ni (to the park) iki-masu (I'll go), anata-no

ani-to (with your brother) kare-no kuruma-de (by his car) tsugi-no nichiyobi-ni (next Sunday) ikimasu. This is equivalent to having a correction as a result.

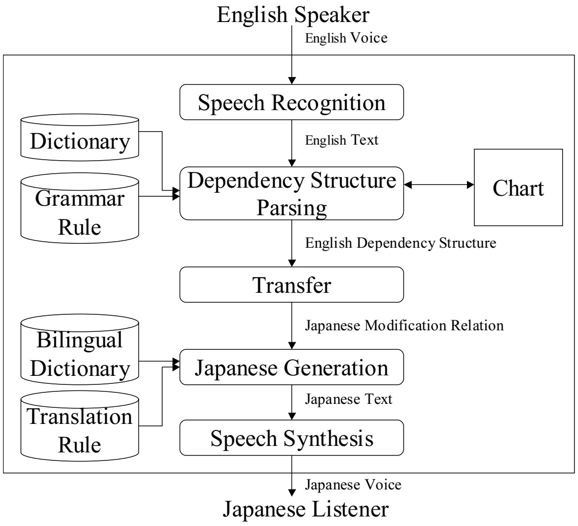

Figure 1: Simultaneous interpreting System *LINAS*

3 English-to-Japanese Simultaneous interpreting system *LINAS*

In this section, we introduce the simultaneous interpreting system *LINAS*, which has two main modules, dependency structure parsing module and Japanese generation module. We'll introduce these modules in more detail.

3.1 Overview of *LINAS*

The protoyepe system of incremental English-to-Japanese tranlstion, *LINAS*, has been developed. Figure 1 shows the overview of *LINAS*. *LINAS* is composed of four language processing modules: speech recognition, incremental parsing, transfer, Japanese generation and speech synthesis. The incremental parsing constructs the dependency structures on a word-by-word bassis for English spoken sentence. Efficient dependency structure analysis is accomplished by utilizing the reachability relations between categories (Y. Kato, 2001). The transfer transforms from English dependency structure to Japanese modification relations and carries out lexical conversion. Moreover, Japanese generation makes Japanese translation. In this paper, we change the Japanese generation utilizing inversions.

3.2 Incremental Parsing

LINAS uses an incremental chart parsing (S. Matsubara, 1997) to parse. The incremental chart parsing, which is one of the syntax analysis techniquer using a context free grammar, expresses the result in progress as the graph called **chart** (M. Kay, 1980). That is, the incremental chart parsing introduces two new operationns into the standard bottom-up chart parsing. One is the application of grammar rules to an active edge and the other is the replacement of the leftmost undecided term with the term of an active edge. In the incremental chart parsing, when i-th word w_i is produced, the following procedures are performed in order:

1)Consultation of a dictionary :
If the category of a word w_i is A, the edge labeled term $[w_i]_A$ is stretched between nodes $i-1$ and i in a chart.

2)Application of grammar rules :
If the term of the edge stretched between nodes $i-1$ and i in a chart is $[\cdots]_{A_1}$ and a grammar rule $A \rightarrow A_1 A_2 \cdots A_n$ exsts, the edge labeled term $[[\cdots]_{A_1}[?]_{A_2} \cdots [?]_{A_n}]_A$ is added between nodes $i-1$ and i in a chart. As much as possible, this operation is repeated.

3)Replacement of terms :
Let $[?]_X$ be the leftmost undecided term of the term σ of the edge stretched between nodes 0 and $i-1$ in a chart. If the category of the term τ of the edge stereched between $i-1$ and i is X, the edge labeled the term produced by replacing the leftmost undecided term of σ with τ is added between nodes 0 and i in a chart.

In order to map from a syntactic structure to English dependency relations, every grammar rules are attached a **head** word and **implement** words. We can get English dependency relations by using these rules.

3.3 Procedures of Generation Utiliaing No Inversion

In this section, the procedure for generating a Japanese translation including no inversion is shown. When a word is inputed, the following procedures are performed in order:

1)Dicides whether a word can be outputed :
It dicides that input words can be outputed if either following condition is fulfilled.

Condition 1 :No word modifies the word

Condition 2 :All words modifying the word have already been outputed or can be outputed.

This continue until no word is judged that it can be outputed.

2)Dicision of the word order of a translation :
In order to fulfill dependency constraints in Japanese it orders the tranlation words which can be outputed.

3)Output :
It outputs a translation ordered in Step2 and changes the state of the words outputed "output".

4)Correction caused by parsing error :
If a word have already been outputed and the other word modifying it have not been outputed, then it changes the state of it "Not-output"and corrects.

4 Method of Generation Utilizing Inversion

In this section, the procedure for generating a Japanese translation including inversion is shown.

4.1 Procedures of Generation Utilizing Inversion

When a word is inputed, the following procedures are performed in order:

1) Dicides whether a word can be outputed :
It dicides that input words except predicate can be outputed if either following condition is fulfilled.

Condition 1 :No word modifies the word

Condition 2 :All words modifying the word have already been outputed or can be outputed.

This continue until no word is judged that it can be outputed.

2)Dicision of the word order of a translation :
In order to fulfill dependency constraints in Japanese it orders the tranlation words which can be outputed. If a predicate has already been outputed, then go to Step 3.1 else go to Step 3.2.

3.1)Dicides whether predicate can be outputed :
It dicides that a predicate can be outputed, if conditions of the generation of a predicate is fulfilled.

3.2)Correction caused by inversions :
If the features of inversions described in Section 2.2 is not fulfilled, then it changes the state of predicate "Not-output" and corrects.

4)Output :
It outputs the translation ordered in Step2 and changes the state of outputed words "output".

5)Correction caused by parsing error :
If a word have already been outputed and the other word modifying it has not been outputed, then it changes the state of the word "Not-output"and corrects.

4.2 Example of Generation

Table 1 shows the system processing when the sentence "I prepare the room with the bath for you." is inputed. The parsing uses the following grammar rules. $\mathrm{S} \rightarrow \mathrm{S}_0 \quad \$^*$

$\mathrm{S}_0 \rightarrow \quad \mathrm{NP}_{subj} \quad \mathrm{VP}^*$
$\mathrm{NP} \rightarrow \quad \mathrm{PRON}^* \;\mid\; \mathrm{DET} \quad \mathrm{N}^* \;\mid\; \mathrm{DET} \quad \mathrm{N}^* \quad \mathrm{PP}$
$\mathrm{VP} \rightarrow \quad \mathrm{V}^* \quad \mathrm{NP}_{obj} \quad \mathrm{PP}$
$\mathrm{PP} \rightarrow \quad \mathrm{P}^* \quad \mathrm{NP}$
PRON→ I | you
N→ room | bath
V→ prepare
DET→ the
$ → . The above rules have one special category called **head**, which is indicated by the symbol "*". The other categories in right-hand side of rules are called **complement**. And the **case** information, subject and object, is indicated by the subindex of the category symbols.

5 Experiment

A English-to-Japanese translation experiment was made on *LINAS* and the effectiveness of our

Table 2: Experimental result

Method	delay	Quality of Translation		
		Good	Acceptable	Unacceptable
Old Method	2.65	18	8	46
New Method	2.03	17	7	48

technique utilizing inversions was evaluated. In the experiment, 578 sentences of ATIS corpus were used, and the delay and the quality of translations was compared between Japanese Generation using inversions and that using no inversion. To evaluate the delay we examined an average delay time of translation words in these generation methods. We used the following expression to compute the average delay time D.

$$D = \sum_{k=1}^{n} \frac{d_k}{n} \qquad (1)$$

In the above expression, d_k is an elapsed time from input the kth word until output its translation word. (When a correction is used, d_k is until restate its translation word.) An elapsed time goes up one unit time whenever a word is inputed. And n is the number of words of English sentence. We evaluated the quality of translations in three stages, good, acceptable and unacceptable, manually. Figure 2 showns the result of the experiment. As a result of the experiment, Inversions were used in 72 translations among 578 translations and Corrections were used in 12 translations. Moreover the delay time decreased an average of about 23% per a word and the quality of translations decreased about only 2.8%. So the effect of this technique can be confirmed by the experiment.

6 Conclusion

In this paper, we proposed the method of a Japanese generation utilizing inversions so that the simultanesous interpreting system had been advanced in respect of the simultaneity of output. The English-to-Japanese translation experiment was made on *LINAS* and the effectiveness of our technique utilizing inversions was evaluated. As a result of the experiment, the improvement in the simultaneity and the maintenance in the quality can be confirmed in the experimnet.

References

J. W. Amtrup. 1999. Incremental Speech Translation. *Lecture Notes in Artificial Intelligence*, 1735.

Y. Kato, S. Matsubara, K. Toyama and Y. Inagaki. 2001. Efficient Incremental Dependency Parsing. *Proceedings of 6th International Workshop of Parsing Technology*, pp.225-228.

M. Kay 1980. Algorithm Schemata and Data Structure in Syntactic Processing. *Technical Report* CSL-80-12.

M. P. Marcus, B. Santorini and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: Penn treebank. *Computational Linguistics*, 19(2).

S. Matsubara, and Y. Inagaki. 1997. Incremental Transfer in English-Japanese Machine Translation. *IEICE Transactions on Information and System*, Vol. E80-D, No. 11, pp. 1122-1129.

H. Mima, H. Iida, O. Furuse. 1998. Simultaneous Interpretation Utilizing Example-based Incremental Transfer. *Proceedings of 15th International Conference on Computational Linguistics.* pp. 858-861.

T. Ohno, S. Matsubara, N. Kawaguchi, and Y. Inagaki. 2003 Spiral Construction of Syntactically Annotated Spoken Language Corpus. *Proceedings of 2003 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pp. 477-483.

Table 1: Example of interpretation process in *LINAS*

Input	Parse Tree	Modification Relation	No Inversion		Inversions	
			Output Possible	Output	Output Possible	Output
I	$[[[[I]_{pron}]_{np}[?]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, ? >$				
prepare	$[[[[I]_{pron}]_{np}[[prepare]_{v}[?]_{np}[?]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, ?, prepare >$ $< null, ?, prepare >$ $< null, prepare, ? >$	I	watashi-ha	I	watashi-ha
the	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[?]_{n}[?]_{pp}]_{np}[?]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, ?, prepare >$ $<?, ?, prepare >$ $< null, the, ? >$ $< null, prepare, ? >$				
room	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[?]_{pp}]_{np}[?]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $<?, ?, prepare >$ $< null, the, room >$ $< null, prepare, ? >$	the room	heya-wo	the room	heya-wo
with	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[[with]_{p}[?]_{np}]_{pp}]_{np}[?]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $< with, with, prepare >$ $< null, the, room >$ $< null, ?, with >$ $< null, prepare, ? >$		eeto		eeto
the	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[[with]_{p}[[the]_{det}[?]_{n}]_{np}]_{pp}]_{np}[?]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $< with, with, prepare >$ $< null, the, room >$ $< null, ?, with >$ $< null, the, ? >$ $< null, prepare, ? >$				
bath	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[[with]_{p}[[the]_{det}[bath]_{n}]_{np}]_{pp}]_{np}[?]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $<?, ?, prepare >$ $< null, the, room >$ $< with, with, room >$ $< null, bath, with >$ $< null, the, bath >$ $< null, prepare, ? >$	the bath with the room	yokushitsu no-aru heya-wo	the bath with the room prepare	yokushitsu no-aru heya-wo jyunbi-shimasu
for	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[[with]_{p}[[the]_{det}[bath]_{n}]_{np}]_{pp}]_{np}[[for]_{p}[?]_{np}]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $< for, for, prepare >$ $< null, the, room >$ $< with, with, room >$ $< null, bath, with >$ $< null, the, bath >$ $< null, ?, for >$ $< null, prepare, ? >$				
you	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[[with]_{p}[[the]_{det}[bath]_{n}]_{np}]_{pp}]_{np}[[for]_{p}[[you]_{pron}]_{np}]_{pp}]_{vp}]_{s_0}[?]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $< for, for, prepare >$ $< null, the, room >$ $< with, with, room >$ $< null, bath, with >$ $< null, the, bath >$ $< null, you, for >$ $< null, prepare, ? >$	you for	anata no-tameni	you for	anata no-tameni
.	$[[[[I]_{pron}]_{np}[[prepare]_{v}[[the]_{det}[room]_{n}[[with]_{p}[[the]_{det}[bath]_{n}]_{np}]_{pp}]_{np}[[for]_{p}[[you]_{pron}]_{np}]_{pp}]_{vp}]_{s_0}[.]_{\$}]_{s}$	$< subj, I, prepare >$ $< obj, room, prepare >$ $< for, for, prepare >$ $< null, the, room >$ $< with, with, room >$ $< null, bath, with >$ $< null, the, bath >$ $< null, you, for >$ $< null, prepare, . >$	prepare	jyunbi-shimasu		