Simultaneous English-Japanese Spoken Language Translation Based on Incremental Dependency Parsing and Transfer

Koichiro Ryu
Graduate School of
Information Science,
Nagoya University
Furo-cho, Chikusa-ku,
Nagoya, 464-8601, Japan
ryu@el.itc.nagoya-u.ac.jp

Shigeki Matsubara
Information Technology Center,
Nagoya University
Furo-cho, Chikusa-ku,
Nagoya, 464-8601, Japan

Yasuyoshi Inagaki
Faculty of
Information Science
and Technology,
Aichi Prefectural University
Nagakute-cho, Aichi-gun,
Aichi-ken, 480-1198, Japan

Abstract

This paper proposes a method for incrementally translating English spoken language into Japanese. To realize simultaneous translation between languages with different word order, such as English and Japanese, our method utilizes the feature that the word order of a target language is flexible. To resolve the problem of generating a grammatically incorrect sentence, our method uses dependency structures and Japanese dependency constraints to determine the word order of a translation. Moreover, by considering the fact that the inversion of predicate expressions occurs more frequently in Japanese spoken language, our method takes advantage of a predicate inversion to resolve the problem that Japanese has the predicate at the end of a sentence. Furthermore, our method includes the function of canceling an inversion by restating a predicate when the translation is incomprehensible due to the inversion. We implement a prototype translation system and conduct an experiment with all 578 sentences in the ATIS corpus. The results indicate improvements in comparison to two other methods.

1 Introduction

Recently, speech-to-speech translation has become one of the important research topics in machine translation. Projects concerning speech translation such as TC-STAR (Hoge, 2002) and DARPA Babylon have been executed, and conferences on spoken language translation such as IWSLT have been held. Though some speech translation systems have been developed so far (Frederking et al., 2002; Isotani et al., 2003; Liu et al., 2003; Takezawa et al., 1998), these systems, because of their sentence-by-sentence translation, cannot start to translate a sentence until it has been fully uttered. The following problems may arise in cross-language communication:

- The conversation time become long since it takes much time to translate
- The listener has to wait for the translation since such systems increase the difference between the beginning time of the speaker's utterance and the beginning time of its translation

These problems are likely to cause some awkwardness in conversations. One effective method of improving these problems is that a translation system begins to translate the words without waiting for the end of the speaker's utterance, much as a simultaneous interpreter does. This has been verified as possible by a study on comparing simultaneous interpretation with consecutive interpretation from the viewpoint of efficiency and smoothness of cross-language conversations (Ohara et al., 2003).

There has also been some research on simultaneous machine interpretation with the aim of developing environments that support multilingual communication (Mima et al., 1998; Casacuberta et al., 2002; Matsubara and Inagaki, 1997).

To realize simultaneous translation between languages with different word order, such as English and Japanese, our method utilizes the feature that the word order of a target language is flexible. To resolve the problem that translation systems generates grammatically dubious sentence, our method utilizes dependency structures and Japanese dependency constraints to determine the word order of a translation. Moreover, by considering the fact that the inversion of predicate expressions occurs more frequently in Japanese spoken language, our method employs predicate inversion to resolve the problem that Japanese has the predicate at the end of the sentence. Furthermore, our method features the function of canceling an inversion by restating a predicate when the translation is incomprehensible due to the inversion. In the research described in this paper, we implement a prototype translation system, and to evaluate it, we conduct an experiment with all 578 sentences in the ATIS corpus.

This paper is organized as follows: Section 2 discusses an important problem in English-Japanese simultaneous translation and explains the idea of utilizing flexible word order. Section 3 introduces our method for the generation in English-Japanese simultaneous translation, and Section 4 describes the configuration of our system. Section 5 reports the experimental results, and the paper concludes in Section 6.

2 Japanese in Simultaneous English-Japanese Translation

In this section, we describe the problem of the difference of word order between English and Japanese in incremental English-Japanese translation. In addition, we outline an approach of simultaneous machine translation utilizing linguistic phenomena, flexible word order, and inversion, characterizing Japanese speech.

2.1 Difference of Word Order between English and Japanese

Let us consider the following English:

(E1) I want to fly from San Francisco to Denver next Monday.

The standard Japanese for (E1) is

(J1) raishu-no (*'next'*) getsuyobi-ni (*'Monday'*) San Francisco-kara (*'from'*) Denver-he (*'to'*) tobi-tai-to omoi-masu (*'want to fly'*).

Figure 1 shows the output timing when the translation is generated as incrementally as possible in consideration of the word alignments between (E1) and (J1). In Fig. 1, the flow of time is shown from top to bottom. In this study, we assume that the system translates input words chunk-by-chunk. We define a simple noun phrase (e.g. San Francisco, Denver and next Monday), a predicate (e.g. want to fly) and each other word (e.g. I, from, to) as a chunk. There is "raishu-no getsuyobi-ni" (*'next Monday'*) at the beginning of the translation (J1), and there is "next Monday" corresponding to "raishu-no getsuyobi-ni" at the end of the sentence (E1). Thus, the system cannot output "raishu-no getsuyobi-ni" and its following translation until the whole sentence is uttered. This is a fatal flaw in incremental English-Japanese translation because there exists an essential difference between English and Japanese in the word order. It is fundamentally impossible to cancel these problems as long as we assume (J1) to be the translation of (E1).

Input	Output
I	
want to fly	
from	
San Francisco	
to	
Denver	
next Monday	raishu-no (*'next'*) getsuyobi-ni (*'Monday'*)
	San Francisco-kara (*'from'*)
	Denver-he (*'to'*)
	tobi-tai-to omoi-masu (*'want to fly'*)

Figure 1: The output timing of the translation (J1)

Input	Output
I	
want to fly	
from	
San Francisco	San Francisco-kara (*'from'*)
to	
Denver	Denver-he (*'to'*) tobi-tai-to omoi-masu (*'want to fly'*)
next Monday	raishu-no (*'next'*) getsuyobi-ni (*'Monday'*)

Figure 2: The output timing of the translation (J2)

2.2 Utilizing Flexible Word Order in Japanese

Japanese is a language with a relatively flexible word order. Thus, it is possible that a Japanese translation can be accepted even if it keeps the word order of an English sentence. Let us consider the following Japanese:

(J2) San Francisco-kara (*'from'*) Denver-he (*'to'*) tobi-tai-to omoi-masu (*'want to fly'*) raishu-no (*'next'*) getsuyobi-ni (*'Monday'*).

(J2) can be accepted as the translation of the sentence (E1) and still keep the word order as close as possible to the sentence (E1). Figure 2 shows the output timing when the translation is generated as incrementally as possible in consideration of the word alignments between (E1) and (J2). The figure demonstrates that a translation system might be able to output “San Francisco -kara (‘*from*’)” when “San Francisco” is input and “Denver-he (‘*to*’) tobi-tai-to omoi-masu (‘*want to fly*’)” when “Denver” is input. If a translation system outputs the sentence (J2) as the translation of the sentence (E1), the system can translate it incrementally. The translation (J2) is not necessarily an ideal translation because its word order differs from that of the standard translation and it has an inverted sentence structure. However the translation (J2) can be easily understood due to the high flexibility of word order in Japanese. Moreover, in spoken language machine translation, the high degree of incrementality is preferred to that of quality. Therefore, our study positively utilizes flexible word order and inversion to realize incremental English-Japanese translation while keeping the translation quality acceptable.

3 Japanese Generation based on Dependency Structure

When an English-Japanese translation system incrementally translates an input sentence by utilizing flexible word order and inversion, it is possible that the system will generate a grammatically incorrect Japanese sentence. Therefore, it is necessary for the system to generate the translation while maintaining the translation quality at an acceptable level as a correct Japanese sentence. In this section, we describe how to generate an English-Japanese translation that retains the word order of the input sentence as much as possible while keeping the quality acceptable.

3.1 Dependency Grammar in English and Japanese

Dependency grammar illustrates the syntactic structure of a sentence by linking individual words. In each link, modifiers (dependents) are connected to the word that they modify (head). In Japanese, the dependency structure is usually defined in terms of the relation between phrasal units called *bunsetsu*[1]. The Japanese dependency relations are satisfied with the following constraints (Kurohashi and Nagao, 1997):

- No dependency is directed from right to left.
- Dependencies do not cross each other.

[1]A *bunsetsu* is one of the linguistic units in Japanese, and roughly corresponds to a basic phrase in English. A bunsetsu consists of one independent word and more than zero ancillary words. A dependency is a modification relation between two bunsetsus.

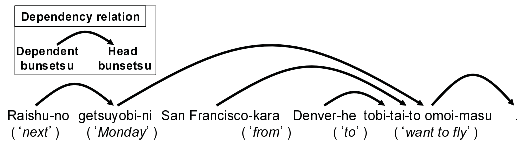


Figure 3: The dependency structures of translation (J1)

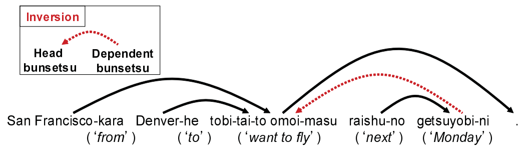


Figure 4: The dependency structures of translation (J2)

- Each bunsetsu, except the last one, depends on only one bunsetsu.

The translation (J1) is satisfied with these constraints as shown in Fig. 3. A sentence satisfying these constraints is deemed grammatically correct sentence in Japanese. To meet this requirement, our method parses the dependency relations between input chunks and generates a translation satisfying Japanese dependency constraints.

3.2 Inversion

In this paper, we call the dependency relations heading from right to left ”inversions”. Inversions occur more frequently in spontaneous speech than in written text in Japanese. That is to say, there are some sentences in Japanese spoken language that do not satisfy the constraint mentioned above. Translation (J2) does not satisfy this constraint, as shown in Fig. 4. We investigated the inversions using the CIAIR corpus (Ohno et al., 2003) and found the following features:

Feature 1 92.2% of the inversions are that the head bunsetsu of the dependency relation is a predicate. (predicate inversion)

Feature 2 The more the number of dependency relations that depend on a predicate increases, the more the frequency of predicate inversions increases.

Feature 3 There are not three or more inversions in a sentence.

From Feature 1, our method utilizes a predicate inversion to retain the word order of an input sentence. It also generates a predicate when the number of dependency relations that depend on a predicate exceeds the constant R (from Feature 2). If there are three or more inversions in the translation, the system cancels an inversion by restating a predicate (from Feature 3).

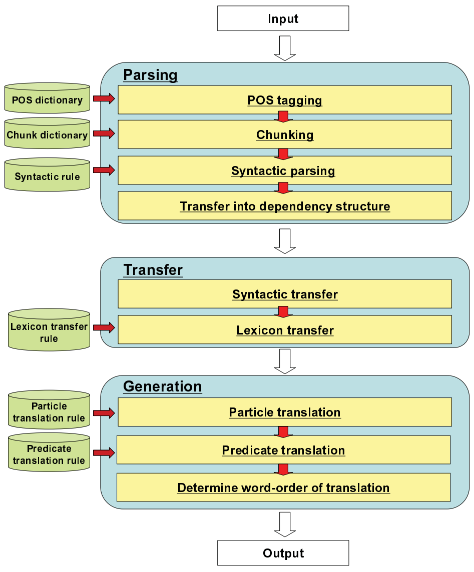


Figure 5: Configuration of our system

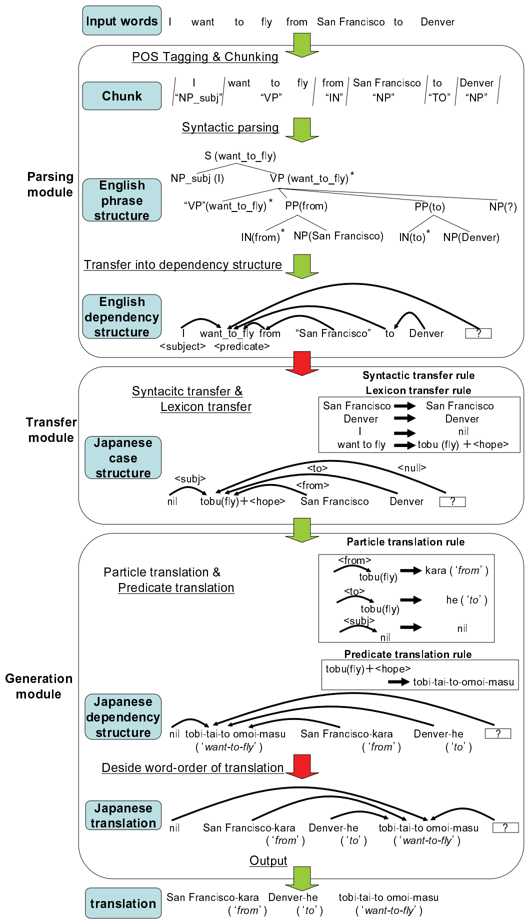
4 System Configuration

Figure 5 shows the configuration of our system. The system translates an English speech transcript into Japanese incrementally. It is composed of three modules: incremental parsing, transfer and generation. In the parsing module the parser determines the English dependency structure for input words incrementally. In the transfer module, structure and lexicon transfer rules transform the English dependency structure into the Japanese case structure. As for the generation module, the system judges whether the translation of each chunk can be output, and if so, outputs the translation of the chunk. Figure 6 shows the processing flow when the fragment “I want to fly from San Francisco to Denver” of （2.1） is input. In the following subsections we explain each module, referring to Fig. 6.

4.1 Incremental Dependency Parsing

First, the system performs POS tagging for input words and chunking (c.f. “Chunk” in Fig. 6).

Next, we explain how to parse the English phrase structure (c.f. “English phrase structure” in Fig. 6). When we parse the phrase structure for input words incrementally, there arises the problem of ambiguity; our method needs to determine only one parsing result at a time. To resolve this problem our system selects the phrase structure of the maximum likelihood at that time by using PCFG (Probabilistic Context-Free Grammar) rules. To resolve the problem of the processing time our system sets a cut-off value.



Input words
I want to fly from San Francisco to Denver
POS Tagging & Chunking
Chunk
I "NP_subj"
want to fly "VP"
from "IN"
San Francisco "NP"
to "TO"
Denver "NP"
Syntactic parsing
Parsing module
English phrase structure
S (want_to_fly)
NP_subj (I)
VP (want_to_fly)*
"VP"(want_to_fly)*
PP(from)
PP(to)
NP(?)
IN(from)*
NP(San Francisco)
IN(to)*
NP(Denver)
Transfer into dependency structure
English dependency structure
I want_to_fly from "San Francisco" to Denver ?
<subject> <predicate>
Transfer module
Syntacitc transfer & Lexicon transfer
Syntactic transfer rule
Lexicon transfer rule
San Francisco → San Francisco
Denver → Denver
I → nil
want to fly → tobu (fly) + <hope>
Japanese case structure
<subj>
<to>
<null>
<from>
nil tobu(fly)+<hope> San Francisco Denver ?
Generation module
Particle translation & Predicate translation
Particle translation rule
<from> tobu(fly) → kara ('from')
<to> tobu(fly) → he ('to')
<subj> nil → nil
Predicate translation rule
tobu(fly)+<hope> → tobi-tai-to omoi-masu
Japanese dependency structure
nil tobi-tai-to omoi-masu ('want-to-fly') San Francisco-kara ('from') Denver-he ('to') ?
Deside word-order of translation
Japanese translation
nil San Francisco-kara ('from') Denver-he ('to') tobi-tai-to omoi-masu ('want-to-fly') ?
Output
translation
San Francisco-kara ('from') Denver-he ('to') tobi-tai-to omoi-masu ('want-to-fly')

Figure 6: The translation flow for the fragment “I want to fly from San Francisco to Denver”

Furthermore, the system transforms the English phrase structure into an English dependency structure (c.f. “English dependency structure” in Fig. 6). The dependency structure for the sentence can be computed from the phrase structure for the input words by defining the category for each rule in CFG, called a ”head child” (Collins, 1999). The head is indicated using an asterisk * in the phrase structure of Fig. 6. In the “English phrase structure,” the chunk in parentheses at each node is the head chunk of the node that is determined by the head information of the syntax rules. If the head chunk (e.g. “from”) of a child node (e.g. PP(from)) differs from that of its parent node (e.g. VP(want-to-fly)), the head chunk (e.g. “from”) of the child node depends on the head chunk (e.g. “want-to-fly”) of the parent node. Some syntax rules are also annotated with subject and object information. Our system uses such information to add Japanese function words to the translation of the subject chunk or the object chunk in the generation module. To use a predicate inversion in the generation module the system has to recognize the predicate of an input sentence. This system recognizes the chunk (e.g. "want to fly") on which the subject chunk (e.g. "I") depends as a predicate.

4.2 Incremental Transfer

In the transfer module, structure and lexicon transfer rules transform the English dependency structure into the Japanese case structure ("Japanese case structure" in Fig. 6). In the structure transfer, the system adds a type of relation to each dependency relation according to the following rules.

- If the dependent chunk of a dependency relation is a subject or object (e.g. "I"), then the type of such dependency relation is "subj" or "obj".
- If a chunk A (e.g. "San Francisco") indirectly depends on another chunk B (e.g. "want-to-fly") through a preposition (e.g. "from"), then the system creates a new dependency relation where A depends on B directly, and the type of the relation is the preposition.
- The type of the other relations is "null".

In the lexicon transfer, the system transforms each English chunk into its Japanese translation.

4.3 Incremental Generation

In the generation module, the system transforms the Japanese case structure into the Japanese dependency structure by translating a particle and a predicate. In attaching a particle (e.g. "kara" (from)) to the translation of a chunk (e.g. "San Francisco"), the system determines the attached particle (e.g. "kara" (from)) by particle translation rules. In translating a predicate (e.g. "want to fly"), the system translates a predicate by predicate translation rules, and outputs the translation of each chunk using the method described in Section 3.

4.4 Example of Translation Process

Figure 7 shows the processing flow for the English sentence, "I want to fly from San Francisco to Denver next Monday." In Fig. 7 the underlined words indicate that they can be output at that time.

5 Experiment

5.1 Outline of Experiment

To evaluate our method, we conducted a translation experiment was made as follows. We implemented the system in Java language on a 1.0-GHz PentiumM PC with 512 MB of RAM. The OS was Windows XP. The experiment used all 578 sentences in the ATIS corpus with a parse tree, in the Penn Treebank (Marcus et al. 1993). In addition, we used 533 syntax rules, which were extracted from the corpus' parse tree. The position of the head child in the grammatical rule was defined according to Collins' method (Collins, 1999).

5.2 Evaluation Metric

Since an incremental translation system for spoken dialogues is required to realize a quick and informative response to support smooth communication, we evaluated the translation results of our system in terms of both simultaneity and quality.

To evaluate the translation quality of our system, each translation result of our system was assigned one of four ranks for translation quality by a human translator:

A (Perfect): no problems in either information or grammar

B (Fair): easy to understand but some important information is missing or it is grammatically flawed

C (Acceptable): broken but understandable with effort

D (Nonsense): important information has been translated incorrectly

To evaluate the simultaneity of our system, we calculated the average delay time for translating chunks using the following expression:

$$Average\ delay\ time = \frac{\sum_k d_k}{n}, \qquad (1)$$

where d_k is the virtual elapsed time from inputting the kth chunk until outputting its translated chunk. (When a repetition is used, d_k is the elapsed time from inputting the kth chunk until restate its translated chunk.) The virtual elapsed time increases by one unit of time whenever a chunk is input, n is the total number of chunks in all of the test sentences.

The average delay time is effective for evaluating the simultaneity of translation. However, it is difficult to evaluate whether our system actually improves the efficiency of a conversation. To do so, we measured "the speaker' and the interpreter's utterance time." "The speaker' and the interpreter 'utterance time" runs from the start time of a speaker's utterance to the end time of its translation. We cannot actually measure actual "the

Table 1: Comparing our method (Y) with two other methods (X, Z)

Method	Quality			Average delay time	Speaker and interpreter utterance time (sec)
	A	A+B	A+B+C		
X	7 (1.2%)	48 (8.3%)	92 (15.9%)	0	4.7
Y	40 (6.9%)	358 (61.9%)	413 (71.5%)	2.79	6.0
Z				3.79	6.4

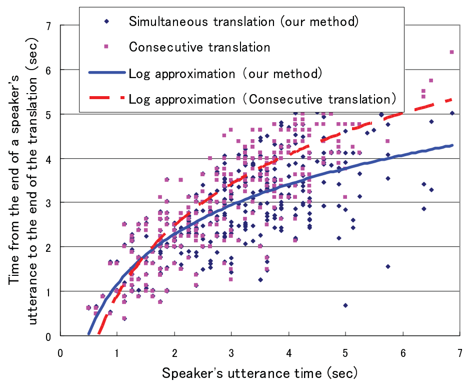


Figure 8: The relation between the speaker's utterance time and the time from the end time of the speaker's utterance to the end time of the translation

speaker' and the interpreter' utterance time" because our system does not include speech recognition and synthesis. Thus, the processing time of speech recognition and transfer text-to-speech synthesis is zero, and the speaker's utterance time and the interpreter's utterance time is calculated virtually by assuming that the speaker's and interpreter's utterance speed is 125 ms per mora.

5.3 Experiment Results

To evaluate the translation quality and simultaneity of our system, we compared the translation results of our method (Y) with two other methods. One method (X) translates the input chunks with no delay time. The other method (Z) translates the input chunks by waiting for the whole sentence to be input, in as consecutive translation. We could not evaluate the translation quality of the method Z because we have not implemented the method Z. And we virtually compute the delay time and the utterance time. Table 1 shows the estimation results of methods X, Y and Z. Note, however, that we virtually calculated the average delay time and the speaker's and interpreter's utterance times in method Z without translating the input sentence.

Table 1 indicates that our method Y achieved a 55.6% improvement over method X in terms of translation quality and a 1.0 improvement over method Z for the average delay time.

Figure 8 shows the relation between the speaker's utterance time and the time from the end time of the speaker's utterance to the end time of the translation. According to Fig. 8, the longer a speaker speaks, the more the system reduces the time from the end time of the speaker's utterance to the end time of the translation.

In Section 3, we explained the constant R. Table 2 shows increases in R from 0 to 4, with the results of the estimation of quality, the average delay time, the number of inverted sentences and the number of sentences with restatement. When we set the constant to $R = 2$, the average delay time improved by a 0.08 over that of method Y, and the translation quality did not decrease remarkably. Note, however, that method Y did not utilize any predicate inversions.

To ascertain the problem with our method, we investigated 165 sentences whose translations were assigned the level D when the system translated them by utilizing dependency constraints. According to the investigation, the system generated grammatically incorrect sentences in the following cases:

- There is an interrogative word (e.g. "what", "which") in the English sentence (64 sentences).
- There are two or more predicates in the English sentence (25 sentences).
- There is a coordinate conjunction (e.g. "and", "or") in the English sentence (21 sentences).

Other cases of decreases in the translation quality occurred when a English sentence was ill-formed or when the system fails to parse.

6 Conclusion

In this paper, we have proposed a method for incrementally translating English spoken language into Japanese. To realize simultaneous translation

Table 2: The results of each R ($0 \leq R \leq 4$)

R	Quality			Average delay time	Sentences with inversion	Sentences with restatement
	A	A+B	A+B+C			
0	8 (1.4%)	152 (26.3%)	363 (62.8%)	2.51	324	27
1	14 (2.4%)	174 (30.1%)	364 (63.0%)	2.53	289	29
2	36 (6.2%)	306 (52.9%)	396 (68.5%)	2.71	73	5
3	39 (6.7%)	344 (59.5%)	412 (71.3%)	2.79	28	2
4	40 (7.0%)	358 (61.9%)	412 (71.3%)	2.79	3	2

our method utilizes the feature that word order is flexible in Japanese, and determines the word order of a translation based on dependency structures and Japanese dependency constraints. Moreover, our method employs predicate inversion and repetition to resolve the problem that Japanese has a predicate at the end of a sentence. We implemented a prototype system and conducted an experiment with 578 sentences in the ATIS corpus. We evaluated the translation results of our system in terms of quality and simultaneity, confirming that our method achieved a 55.6% improvement over the method of translating by retaining the word order of an original with respect to translation quality, and a 1.0 improvement over the method of consecutive translation regarding average delay time.

Acknoledgments

The authors would like to thank Prof. Dr. Toshiki Sakabe. They also thank Yoshiyuki Watanabe, Atsushi Mizuno and translator Sachiko Waki for their contribution to our study.

References

F. Casacuberta, E. Vidal and J. M. Vilar. 2002. Architectures for speech-to-speech translation using finite-state models, *Proceedings of Workshop on Speech-to-Speech Translation: Algorithms and System*, pages 39-44.

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing, *Ph.D. Thesis, University of Pennsylvania*,

R. Frederking, A. Blackk, R. Brow, J. Moody, and E. Stein-brecher, 2002. Field Testing the Tongues Speech-to-Speech Machin Translation System, *Proceedings of the 3rd International Conference on Language Resources and Evaluation(LREC-2002)* pages 160-164.

H. Hoge. 2002. Project Proposal TC-STAR: Make Speech to Speech Translation Real, *Proceedings of the 3rd International Conference on Language Resources and Evaluation(LREC-2002)*, pages 136-141.

R. Isotani, K. Yamada, S. Ando, K. Hanazawa, S. Ishikawa and K. Iso. 2003. Speech-to-Speech Translation Software PDAs for Travel Conversation, *NEC Research and Development, 44, No.2* pages 197-202.

S. Kurohashi and M. Nagao. 1997. Building a Japanese Parsed Corpus while Improving the Parsing System, *Proceedings of 4th Natural Language Processing Pacific Rim Symposium*, pages 451-456.

F. Liu, Y. Gao, L. Gu and M. Picheny. 2003. Noise Robustness in Speech to Speech Translation, *IBM Tech Report RC22874*.

M. P. Marcus, B. Santorini and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank, *Computational Linguistics*, 19(2):310-330.

S. Matsubara and Y. Inagaki. 1997. Incremental Transfer in English-Japanese Machine Translation, *IEICE Transactions on Information and Systems*, (11):1122-1129.

H. Mima, H. Iida and O. Furuse. 1998. Simultaneous Interpretation Utilizing Example-based Incremental Transfer, *Proceedings of 17th International Conference on Computational Linguistics and 36th Annual Meeting of Association for Computational Linguistics*, pages 855-861.

M. Ohara, S. Matsubara, K. Ryu, N. Kawaguchi and Y. Inagaki. 2003. Temporal Features of Cross-Lingual Communication Mediated by Simultaneous Interpreting: An Analysis of Parallel Translation Corpus in Comparison to Consecutive Interpreting, *The Journal of the Japan Association for Interpretation Studies* pages 35-53.

T. Ohno, S. Matsubrara, 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*, pages 477-483.

T. Takezawa, T. Morimoto, Y. Sagisaka, N. Campbell, H. Iida, F. Sugaya, A. Yokoo and S. Yamamoto. 1998. A Japanese-to-English Speech Translation System:ATR-MATRIX, *Proceedings of 5th International Conference on Spoken Language Processing*, pages 957-960.

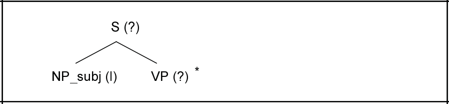
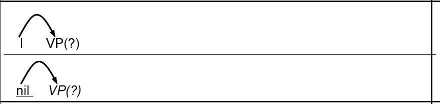
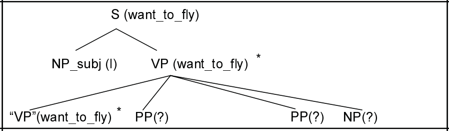
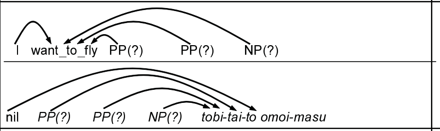
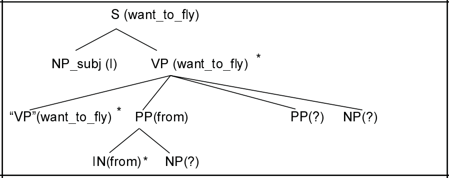
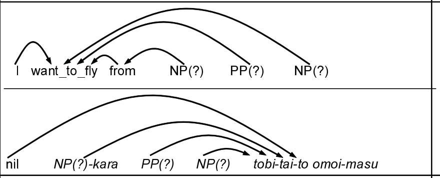
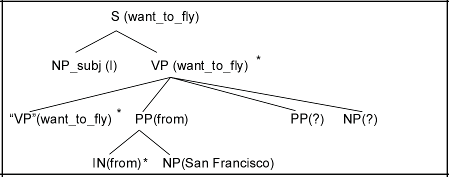
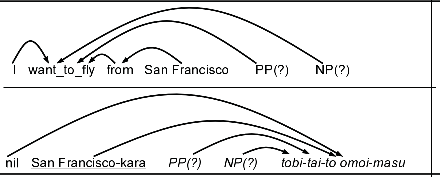
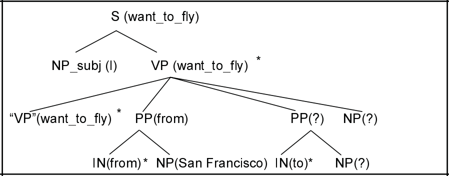
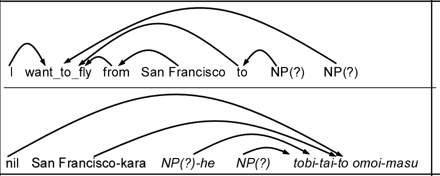
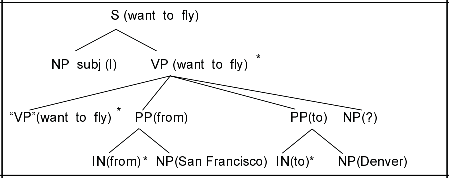
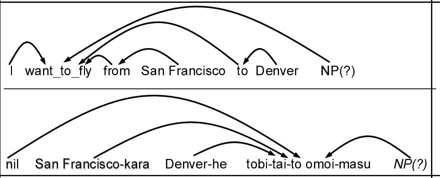
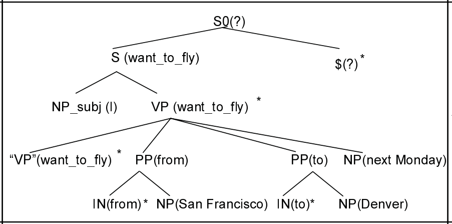
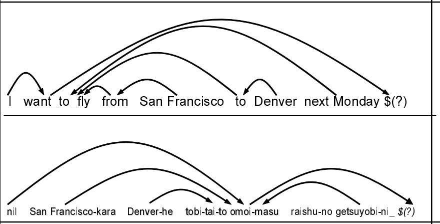
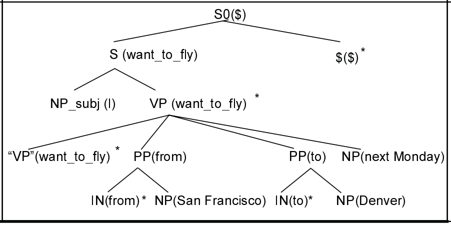
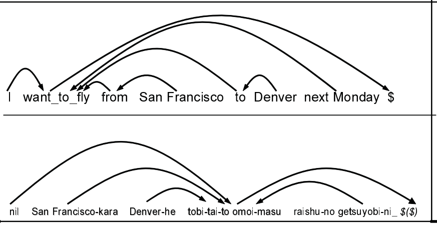
Input	Parse tree	English dependency structure / Japanese dependency structure	Output
I			nil
want to fly			
from			
San Francisco			San Francisco -kara (*'from'*)
to			
Denver			Denver-he (*'to'*) tobi-tai-to omoi-masu (*'want to fly'*)
next Monday			raishu-no (*'next'*) getsuyobi-ni (*'Monday'*)
.			.

Figure 7: The translation flow for "I want to fly from San Francisco to Denver next Monday."