社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

有限状態トランスデューサを用いた話し言葉の同時翻訳

笠　浩一朗† 　松原　茂樹†† 　稲垣　康善†††

† 名古屋大学大学院情報科学研究科 〒 464-8601 名古屋市千種区不老町
†† 名古屋大学情報連携基盤センター 〒 464-8601 名古屋市千種区不老町
††† 愛知県立大学情報科学部 〒 480-1198 愛知県愛知郡長久手町大字熊張字茨ヶ廻間 1522-3
E-mail: †ryu@el.itc.nagoya-u.ac.jp

あらまし　本論文では、対訳コーパスを用いた話し言葉の同時翻訳手法を提案する。本手法は、対訳データから獲得した翻訳パターンを利用するパターンベース翻訳として実現する。対訳データとしては、対訳文とその逐語訳を利用する。逐語訳データに基づく翻訳パターンを用いることにより、出力の同時進行性を備えた訳文の生成が可能となる。本研究では、高速な同時翻訳処理の実現を目指し、翻訳パターンから作成した有限状態トランスデューサを用いて変換を実行するとして本手法を実現した。本手法の実現可能性を確認するために、実対話文を用いて翻訳実験を実施した。実験では名古屋大学 CIAIR 同時通訳データベースを使用した。
キーワード　機械翻訳, 同時通訳, 対訳コーパス, 訳出タイミング

Simultanous Machine Interpretation using Finite State Transducer

Koichiro RYU†, Shigeki MATSUBARA††, and Yasuyoshi INAGAKI†††

† Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464-8601, Japan
†† Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464-8601, Japan
††† Faculty of Information Science and Technology, Aichi Prefectural University, Nagakute-cho, Aichi-gun, Aichi-ken, 480-1198, Japan
E-mail: †ryu@el.itc.nagoya-u.ac.jp

Abstract　In this paper we present a method of simultanoues machine interpretation useing a parallel corpus. It is based on the pattern-base translation and uses translation patterns acquired from a parallel data. By using word-for-word translations as translation data, our method make it possible to generate translations simultaneously. For high speed translations, our method creates Finite State Transducer(FST) from translation patterns on ahead and uses it to translate. In order to confirm our method's effectivity, we have experimented with our method. 本手法の実現可能性を確認するために、実対話文を用いて翻訳実験を実施した。実験では名古屋大学 CIAIR 同時通訳データベースを使用した。
Key words　machine translation, simultaneous interpretation, bilingual corpus, translation timing

1. はじめに

異言語間対話の支援環境の実現を目的として、音声翻訳システムの開発を目指した研究が進められており、すでにいくつかの実験システムが開発されるに至っている。しかし、通常の同一言語間対話に近い、より自然な異言語対話の遂行を可能にするためには、通訳を介することによる時間的影響が小さいことが望ましく、システムが同時翻訳機能を備えていることは重要である。実際、逐次通訳を介した対話と同時通訳を介した対話を比較した分析では、同時通訳により対話の効率及び円滑さが大幅に向上することが確認されている [4]。

同時翻訳技術についてはこれまでに、漸進的な解析・変換・生成から構成される構文変換に基づく手法が提案されており、プロトタイプシステムを用いた対話翻訳実験により、その利用可能性が検証されている [1], [2], [6]。しかしながら、同時翻訳で用いる規則は、翻訳の品質や精度だけでなく、音声入力に対する訳出の同時性を考慮して設計することが重要であり、大量の翻訳規則を開発者があらかじめ人手で作成することは難しいという問題がある。

上述の問題に対して、話し言葉の同時翻訳を実現するための

対訳コーパスの利用方式について検討することは一つの有力な方法である。 現在までに開発されている音声翻訳技術の多くにおいて、コーパスに基づいた方式が採用されており、その効果は数々の翻訳実験により検証されていること (例えば、[13])、また、これまでにいくつかの研究機関において対訳音声の収集及び蓄積が進められており [3], [7]、大規模な音声対訳データベースの利用環境も次第に整いつつあることなどがその理由である。

しかしながら、対訳コーパスを用いた同時翻訳では、通常のコーパスベース音声翻訳では問題とならなかったいくつかの事項について新たに検討する必要がある。 一般の音声翻訳の場合には、入力された音声をどのような訳文として生成すべきかという、いわゆる “how-to-say” が問題となり、質の高い翻訳文を作り上げる技術の開発に力が注がれてきた。 一方、同時翻訳の場合には、“how-to-say” だけでなく、入力に対してどの時点でどの部分を訳出すべきかという、“when-to-say” さらには “what-to-say” の問題もまた重要となるが、このような問題に対する対訳データの効果的な利用法については明らかではない。

本論文では、対訳コーパスを用いた話し言葉の同時翻訳手法を提案する。 本手法は、対訳データから獲得した翻訳パターンを利用するパターンベース翻訳として実現する。 対訳データとしては、対話文とその逐語訳を利用する。 逐語訳とは、できる限り入力文の語順に準拠してオフラインで作成した翻訳文である。逐語訳データに基づく翻訳パターンを用いることにより、出力の同時進行性を備えた訳文の生成が可能となる。 パターンを構成するフレーズごとの対訳対応を考慮することにより、原言語フレーズの入力に対応して目標言語フレーズを出力するタイミングを決定することが可能となる.

本研究では、高速な同時翻訳処理の実現を目指し、翻訳パターンから作成した有限状態トランスデューサを用いて変換を実行するとして本手法を実現した。 本手法の実現可能性を確認するために、実対話文を用いて翻訳実験を実施した。実験では、名古屋大学 CIAIR 同時通訳データベース [10] を使用した。

本論文の構成は以下の通りである。 続く２章では、対訳コーパスを用いた同時翻訳の概要について論じる。 ３章では、有限状態トランスデューサを用いた同時翻訳手法について説明する。４章では、対話翻訳実験による本手法の評価について述べる。

2. 対訳コーパスを用いた同時通訳

コーパスに基づく機械翻訳手法として、用例ベース翻訳 (example-based machine translation)、パターンベース翻訳 (pattern-based machine translation)、統計翻訳 (statistical machine translation) などがある。 本研究では、パターンベース翻訳により話し言葉の同時翻訳を実現する。 パターンベース翻訳では、対訳コーパスなどから対訳パターンを大量に作成し、入力文に対してそれらを組み合わせて適用することにより対応する訳文を出力する。 本節では、同時翻訳におけるパターンベース方式について検討し、本稿で提案する翻訳手法の概要について述べる。

2.1 同時翻訳における対訳パターンの生成と利用

対訳パターンは、原言語パターンと対応する目的言語パターンから構成され、文構造に関する情報が与えられた対訳データから生成することができる。 原言語入力に対しては、原言語パターンとマッチングする対訳パターンを利用する。対訳パターンの組み合わせにより、目的言語を作り上げることができる。

例えば、 “I'd like to reserve a room at a service counter” と 「サービスカウンタで部屋を予約したいのですが」 という対訳データからは、

$$\text{“}S : I'd\ like\ to\ researve\ NP1\ at\ NP2\text{”} \quad (1)$$
$$\Leftrightarrow \text{“}NP2\ \text{で}\ NP1\ \text{を予約したいのですが”}$$

と

$$\text{“}NP :\ a\ room\text{”} \Leftrightarrow \text{“部屋”} \quad (2)$$

$$\text{“}NP :\ a\ N1\ counter\text{”} \Leftrightarrow \text{“}N1\ \text{カウンタ”} \quad (3)$$

という対訳パターンを生成することができる。 “I'd like to reserve a seat at a ticket counter” が入力されれば、上述の対訳パターン、さらには、

$$\text{“}NP :\ a\ seat\text{”} \Leftrightarrow \text{“席”} \quad (4)$$

というパターンを利用することにより、「] チケットカウンタで席を予約したいのですが”」という訳文を生成できる。 パターンベース方式により同時翻訳を実現するときには、対訳パターンの生成と利用の双方において検討すべき事項が存在する。 対訳パターンの生成に関する事項として、

（１） 対訳パターンの生成に使用する対訳データ
同時翻訳では、音声入力に対する訳出タイミングの同時性が重要となるものの、英語・日本語間の翻訳のように、語の生起順序が大きく異なる言語間の翻訳において高い同時性を達成することは難しい。 しかしながら、プロの同時通訳者が様々な方法で訳出を工夫しているように [9]、生成する訳文のスタイルによっては、高い同時性を備えた翻訳処理も可能となる [2]。同時翻訳のための対訳パターンを大量に蓄積する必要があり、その生成においては適切な対訳データを使用する必要がある。

（２） 訳出タイミングに関する情報
対訳パターンは、原言語パターンに対して生成する目的言語パターンについて定めている。 同時性の高い翻訳を実現するためには、出力可能となったフレーズから順次生成することが望ましく、対訳パターンにおいて、目的言語パターンを構成するフレーズや構文範疇を訳出するタイミングを原言語パターンの構成素との対応として明示することが求められる。 例えば、前出の対訳パターンにおいて、目的言語パターンの”$NP1$” を生成するタイミングが原言語パターンの中のどの構成素が入力された時点であるかを記す必要がある.

を指摘することができる。

一方、対訳パターンの利用に関しては、

（1） 対訳パターンに基づく漸進的変換
一般的なパターンベース翻訳では、一文全体に対して最適な対訳パターンを適用し、目的言語構造を作り上げればよい。 一方、同時翻訳の場合には、目的言語フレーズを順次生成するために、入力途中の段階において対訳パターンを適用することになる。 このような漸進的な変換処理を実行するには、原言語の部分フレーズに対する最適な対訳パターンの選択方法、訳出する目的言語の部分フレーズを唯一に決定する方法、などを検討する必要がある。

（2） 同時翻訳を実現する変換の高速化
翻訳処理における高い同時性を実現するには、少なくとも話者の発声速度と同程度の処理速度を達成する必要がある。 上述したような対訳パターンを適用する場合、蓄積されたパターンが多くなるほど可能なパターンの組み合わせも増加するため、処理に要する時間が増大するという問題がある。

といった事項を検討する必要がある。

2.2 パターンベース同時翻訳

本研究では、パターンベース翻訳として同時翻訳を実現する。本手法の特徴は以下のようにまとめられる。

- 対訳パターンの作成に構文木対応付き逐語訳データを用いる。逐語訳データに基づくことにより、同時性の高い翻訳が可能となる。
- 対訳パターンを構成する構文範疇の対応関係をもとに、できる限り早い段階で生成可能な訳出タイミングを計算し、対訳パターンに明示する。
- 対訳パターンを漸進的に適用することにより、入力途中の段階で訳出可能な部分を決定し生成する。
- 対訳パターンを組み合わせることにより、有限状態トランスデューサ (Finite State Transducer:FST) の形式で翻訳規則を作り上げる。トランスデューサ上を入力に応じて辿るだけで、漸進的に訳文を生成できる。

3. 有限状態トランスデューサを用いた同時翻訳手法

本節では、翻訳規則として FST を用いる同時翻訳手法について述べる。 まず 3.1 節で、訳出タイミング付き対訳パターンの獲得について述べ、 次の 3.2 節で、FST の作成について説明する。 3.3 節では、FST を用いた同時翻訳について述べる。

3.1 対訳パターンの獲得

本手法では、逐語訳コーパスを用いて対訳パターンを生成する。 逐語訳データのサンプルを図Xに示す。 本稿における逐語訳 (word-for-word translation) とは、単語や句などを単位とし、原文の語順に従って目的言語文を生成する訳出方式を指している。 言語単位を翻訳単位として使用するため、対訳パターンが獲得し易いとともに、入力と同時的に出力するための訳文として適している。また、翻訳に精通した作業者が作成した逐語訳を利用することにより、エキスパートの訳出技法を活用できるといった利点もある。

逐語訳コーパスとしては、各発話に対して構文木が付与され、かつ、発話を構成する要素間の対応付けが施されたデータを使用する。 本研究では、このような構文対応データがあらかじめアノテーションされているものとする。 構文的対訳対応が与えられた逐語訳データのサンプルを図[?] に示す。左図が英語原文 “I would like to reserve a room with bath on next Sunday” の構文木であり、右図がその逐語訳「私は風呂付の部屋を次の日曜日に予約したいのですが」 の構文木である。 構文範疇としては、名詞 (NN)、形容詞 (ADJ)、副詞 (ADV) 等の基本品詞や 名詞句 (NP)、形容詞句 (ADJP)、副詞句 (ADVP) 等の基本範疇を使用する。 これらはいずれも単独で翻訳単位を構成することができ、対訳パターンの構成素として利用できる。 構文木の節点は、構文範疇とその ID を用いてラベル付けしており、対応する構文木の同一のラベルをルートとする部分木が対応していることを意味する。 図[?] の英語発話における “a room with bath” が $NP1$ でラベル付けされ、それは同じ $NP1$ でラベル付けられた「風呂付きの部屋」に対応することを示している。

対訳パターンは、対訳対応が付与された構文木データから生成することができる。対応する構文木において、同一の構文範疇がラベル付けされた各節点とその子節点とから、対訳パターンを取り出す。 図[?] に図[?] の対訳対応構文木から取り出した対訳パターンを示す。例えば、先の $NP1$ については、英語の “NP : a $NN2$ $ADJP1$” と日本語の “NP : $ADJP1$ $NN2$” が対応するとして、対訳パターン “NP : a $NN2$ $ADJP1$ $->$ $ADJP1$ $NN2$” を作成する。取り出した対訳パターンは、各範疇ごとにまとめて格納する。

このようにして作り出された対訳パターンは、対訳対応付き構文範疇を含んでいる。 目的言語パターンを構成する要素の出力タイミングは、対訳パターンにおける構文範疇間の関係から決定できる。 すなわち、目的言語パターン上の構成要素の訳出は、パターンに記された生起順序を壊すことなく、かつ、構文範疇に相当する表現は対応する原言語が入力されたのちであるとする。 図[?] に図[?] の対訳パターンから作成した訳出タイミング付き対訳パターンを記す。 先の NP に対する対訳パターンでは、目的言語パターンの “ADJP1 NN2” はいずれも原言語パターンの $ADJP$ が入力された時点であることを意味している。

3.2 有限状態トランスデューサの作成

同一の範疇に属する訳出タイミング付き対訳パターンから, 各範疇ごとの対訳パターンを表現するネットワークを作成する。図 4 から作成されるネットワークを図 5 に示す. 図 4 の一つの原言語単語とその単語が入力されたときに訳出可能な目的言語の単語列の組から, 入力アルファベットが原言語単語で、出力アルファベットが訳出可能な目的言語の単語列の遷移を作成する。

原文１：I would like to reserve a room with bath.	訳文１：私は、バス付き部屋を予約したいのですが
原文２：Your room number is 1001.	訳文２：お客様の部屋の番号は、１００１です。
原文３：I want to reserve a single room on May eighteenth	訳文３：私は、シングルの部屋を５月１８日に予約したいのですが

図１　逐語訳データのサンプル

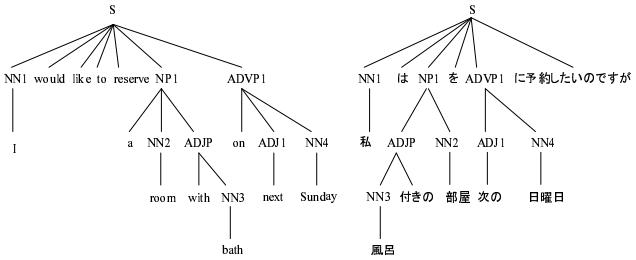

図 2　対訳対応付き対訳例

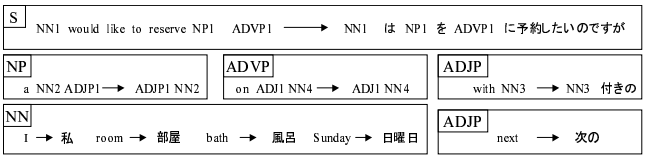

図 3　対訳パターン

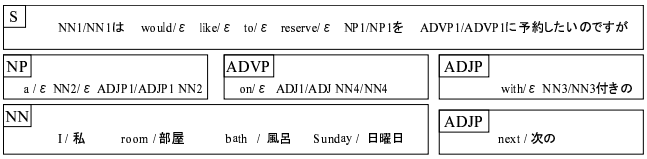

図 4　訳出タイミング付き対訳パターン

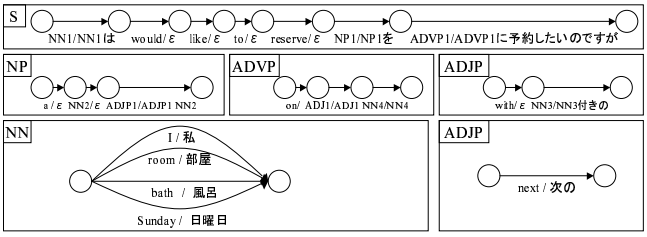

図 5　各範疇のネットワーク

図 5 の各範疇のネットワークから英日同時翻訳に用いる FST を作成する流れを図 6 に示す. まず,S 規則から作成されたネットワークを FST とする. 次に, 初期 FST の入力アルファベットが範疇である部分を, その範疇のネットワークにより展開する. 図の上から二つ目の FST が得られる. さらに同様の範疇の展開を繰り返し行う. 範疇の展開は, 無限に繰り返される可能性があるので, ある一定の制限を決めて展開を行なう. 図 6 の下の FST は展開を 2 回行なったものである。

3.3　有限状態トランスデューサを用いた同時通訳

前節で獲得した FST を用いて同時翻訳する手法について述べる. 基本的には, 入力された原言語の単語を順に FST に入力し, 入力された単語と入力アルファベットが一致する遷移をたどる. また, 最終状態までたどったら, また初期状態に戻り, 入力が終了するまでたどり続ける. ただし, 作成した FST は非決定的であるため, 入力に対して複数の遷移先が存在する可能性がある. そのような場合は, ランダムに遷移を選択し, 遷移先を決定し, その遷移の出力アルファベットを訳として出力する.

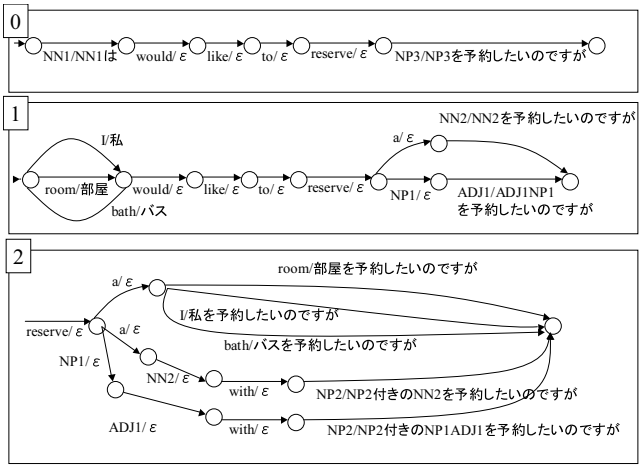

図 6　範疇の展開

表 1　有限状態トランスデューサの学習データ

項目	個数
対訳文数（文）	52
単語総数	389
原文の 1 文あたりの平均単語数	7.48
対訳パターン数	164

表 2　統語レベルの説明と範疇の種類

統語レベル	説明	範疇の種類
語	1 単語からなる単位	名詞, 形容詞, 副詞
句	2 単語以上からなり, 述部を含んでいない単位	名詞句, 形容詞句, 副詞句
節	2 単語以上からなり, 述部を含んでいる単位	名詞節, 形容詞節, 副詞節

4.　翻訳実験

4.1　実験の概要

FST を用いた同時通訳手法の効果を検討するために, 英日翻訳実験を行った. 翻訳実験では、音声認識と音声合成の処理は行なわず、通訳システムへの入力は、入力文を一単語ずつテキストで入力し、また出力は、有限状態変換器から出力される目的言語の単語列とした。実験では、本研究の第一段階としてクローズテストを行った。CIAIR 対話同時通訳コーパスの 1 対話の対訳対応付き対訳例（52 文）から FST を作成し、その学習データの 52 文を入力として用いた。学習データの基礎統計を表 1 に示す。また、対訳例に付与された対訳対応の範疇の種類は、３つの統語レベル（語、句、節）で全 9 種類である (表 2 参照).

FST を作成するときの各範疇を展開する処理において、各範疇ごとに展開して作成した FST には利用される可能性が低い遷移が大量に含まれてしまうので、コーパス中の出現頻度が高い名詞句、名詞をそれぞれ 6 つのカテゴリ (数字、値段、部屋、日付、時間、その他) に分類することで、無駄な遷移を削減した。

FST の構文範疇の展開数は 4 回であり, 各展回数における FST の状態数と遷移数を 3 に示す. 実験で利用した計算機環境

表 3　各展回数での FST の状態数と遷移数

展回数	状態数	遷移数
0	200	253
1	936	1673
2	9264	22176
3	86179	222981
4	794586	2066456

表 4　翻 訳 結 果

項目		文数	
正解文数	発話途中	17	47
	発話終了後	30	
不正解		5	

表 5　翻訳結果（展開数）

展開数	正解文数	不正解文数
0	7	48
1	24	28
2	32	20
3	41	11
4	47	5

は、CPU:Pentium4 2GHz、メモリは 2GB である。

4.2 実 験 結 果

実験結果を表 4 に示す。翻訳結果を主観的に判断して原文の内容が正しく理解できる訳文を正解文とした。翻訳結果より、テスト文 52 文中で正解文となったものは 47 文存在した。その正解文の内、話者の発話の途中段階で翻訳が開始できたものは 17 文存在した。また、1 文あたりの処理速度は約 0.01 秒であり。本手法により高速に翻訳結果を訳出できることがわかった。翻訳失敗した原因について調査したところ、不正解文のすべてが途中で弧を遷移できなくなったためであった。また展回数が 0 回から 4 回までの翻訳結果を表 5 に示す。表 5 より、展開数が増えるごとに正解数が増加していることがわかる。

5. おわりに

本論文では, 対訳対応付き対訳例から作成した対訳パターンを用いて FST を作成することにより、漸進的かつ高速に翻訳する同時通訳手法を提案した.。実験システムを作成し,CIAIR 対話同時通訳コーパスに収録された英語発話 52 文について翻訳実験を行ない, 本手法の有効性を示した.

今後の課題としては, 有限状態トランスデューサには曖昧性が存在するため, 学習データから遷移確率を学習し, 遷移の選択方法を改善する必要がある. また, 同時通訳単位で通訳された訳文間に整合性が取れない場合があるので, 訳文間の整合性を保つ仕組みを提案する必要があると考えている.

謝　　辞

日頃ご指導下さる名古屋大学大学院教授の坂部俊樹先生に感謝致します。 本研究の一部は、文部科学省科学研究費補助金基盤研究 B(2)(課題番号 15300044)、ならびに、(財) 中島記念国際交流財団 研究助成によります。

文　　献

[1] S. Matsubara and Y. Inagaki, “Incremental Transfer in English-Japanese Machine Translation”, *IEICE Transactions on Information and System*, Vol. E80-D, No.11, pp. 1222-1129 (1997).

[2] 松原 茂樹, 浅井 悟, 外山 勝彦, 稲垣 康善: 不適格表現を活用した漸進的な英日話し言葉翻訳手法, 電気学会論文誌, Vol.118-C, No.1, pp.71-78 (1998).

[3] 松原茂樹, 相澤靖之, 河口信夫, 外山勝彦, 稲垣康善: “同時通訳コーパスの設計と構築”, 通訳研究, No.1,pp.85-102 (2001).

[4] 大原誠, 松原茂樹, 笠浩一朗, 河口信夫, 稲垣康善, “同時通訳を介した異言語間対話の時間的特徴” 通訳研究 (日本通訳学会論文誌), No.3, pp.34-52 (2003).

[5] K. Ryu, S. Matsubara, N. Kawaguchi and Y. Inagaki, “Bilingual Speech Dialogue Corpus for Simultaneous Machine Interpretaion Research”, *Proceedings of 6th Oriental-COCOCSDA*, pp.217-224 (2003).

[6] K. Ryu, A. Mizuno, S. Matsubara, and Y. Inagaki: Incremental Japanese Spoken Language Generation in Simultaneous Machine Interpretation, *Proceedings of Asian Symposium on Natural Language Processing to Overcome Language Barriers*, pp. 91-95 (2004).

[7] 竹沢 寿幸, 中村 篤, 隅田 英一郎：ATR の会話音声翻訳研究用データベース, 音声研究, Vol. 4, No. 2, pp. 16-23 (2000).

[8] T. Takezawa, et al., “Japanese-to-English Speech Translation System:ATR-MATRIX”, *Proceedings of 5th International Conference on Spoken Language Processing*, pp. 957-960 (1998).

[9] 遠山 仁美, 松原 茂樹: 同時通訳コーパスを用いた通訳者の訳出パターンの分析,
信学技報 (TL2003-26), pp. 13-18 (2003).

[10] 遠山 仁美, 松原 茂樹, 笠 浩一朗, 河口 信夫, 稲垣 康善: CIAIR 同時通訳データベースの構築と利用, 信学技報, Vol.104, No.170, pp. 7-12 (2004).

[11] T. Watanabe, et al., “An Automatic Interpretation Software for Travel Conversation”, *Proceedings of 6th International Conference on Spoken Language Processing*, vol.IV, pp.444-447 (2000).

[12] T. Watanabe,K.Imamura, and Eiichiro Sumita, “Statistical Machine Translation Based on Hierarchical Phrase Alignment”, *Proceedings of Theoretical and Methodological Issues in Machine Translation*, pp. 188-198 (2002).

[13] 山本誠一：コーパスベース音声翻訳技術、電子情報通信学会誌、Vol. 83, No. 8, pp. 604-611 (2000).