

## 英日同時翻訳のための依存構造に基づく訳文生成手法

笠 浩一朗<sup>†</sup>      松原 茂樹<sup>††</sup>      稲垣 康善<sup>†††</sup>

Japanese Sentence Generation Based on Dependency Structure  
for English-Japanese Simultaneous Translation

Koichiro RYU<sup>†</sup>, Shigeki MATSUBARA<sup>††</sup>, and Yasuyoshi INAGAKI<sup>†††</sup>

あらまし 本論文では、英日同時翻訳のための訳文生成手法を提案する。英語と日本語では、語順が大きく異なるため、標準的な日本語訳を生成したのでは同時性の高い翻訳を実現することは困難である。一方、同時通訳者は、標準的な訳文ではなく、原文の語順に近い語順の日本語文を訳文とすることにより、早い段階での訳出を可能にしている。本手法では、そのような訳出方法を参考に、日本語として容認可能な範囲で、原文の語順に近い語順の日本語訳を入力と同時的に生成する。本論文では、日本語として容認可能な文が満たすべき性質を、依存構造の観点から検討する。また、そのような性質を満たす訳文を生成する手法について述べる。本手法を実装したシステムを用いて翻訳実験を行い、翻訳品質、及び、同時性の観点から本手法を評価した。その結果、容認可能な翻訳品質をもち、かつ、入力に対して同時性が高い英日同時翻訳を実現するために、本手法が有効であることを確認した。

キーワード 機械翻訳、同時通訳、話し言葉処理、依存文法、倒置

### 1. ま え が き

近年の音声処理技術の発展に伴い、音声翻訳は機械翻訳における重要な研究テーマになりつつある。従来の音声翻訳システムの多くは、ターンや文を単位とした翻訳処理が実現されており、話者の発話の終了を待ってから翻訳を開始するものとして実現されている [1]~[4]。このような音声翻訳システムを介して対話を遂行すると、話者発話時間とは別に翻訳処理時間や訳出時間がかかるため、全体の対話時間が増大する。また、聞き手にとっては、話者が発話を開始してから翻訳が開始されるまでが待ち時間となるため、それが対話の円滑さを妨げる要因となる。より自然な対話を可能にするために、同時通訳者のように話者の発話の終了を待たずに翻訳を開始することは一つの方法であ

る [5]。実際、人間の通訳者を介した異言語間対話では、逐次通訳を使用した場合に比べ、同時通訳の方が対話の効率と円滑さが大幅に向上することが確認されている [6]。

同時通訳では、英語と日本語のように、互いに違った性質をもつ言語間の場合には、対応する語の生起順序が異なり、訳出タイミングにおいて困難な問題を引き起こす。例えば、英語は SVO 構文であるのに対し、日本語は SOV 構文であるため、英語話者発話の O の部分の発話が完了するまで V の訳出を待たなければならない。ほかにも、英語では時間を表す副詞が一般に文末に生起するのに対して、日本語では文頭に生起する傾向があるなどの違いが原因となる [7]。

しかしながら、実際の同時通訳者は、様々な訳出方略を駆使することにより上述の問題を克服していると考えられる。英日同時通訳における方略の一つは、英語文の語の生起順序に近い語順で日本語訳を生成することである。同時的な翻訳処理に関する研究はそれほどなく、語順が近い言語間における同時的な変換方法や [8], [9], 日英翻訳において節ごとに変換するための規則及びその適用法 [10] が提案されている段階にあり、上述のような訳出方略を実現するための方法論につい

<sup>†</sup> 名古屋大学大学院国際開発研究科, 名古屋市  
Graduate School of International Development, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464-8601 Japan

<sup>††</sup> 名古屋大学情報基盤センター, 名古屋市  
Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464-8601 Japan

<sup>†††</sup> 豊橋技術科学大学, 豊橋市  
Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku-cho, Toyohashi-shi, 441-8580 Japan

てはこれまで提示されていない。

そこで本論文では、英日同時通訳者による訳出例を参考に、英語対話文の入力に対してできる限り同時進行的に日本語訳文を生成するための手法を提案する。本手法では、日本語の語順に関する制約は比較的緩やかであるという特徴を利用し、日本語文として容認可能な範囲内で、原文の語順に準拠した訳文を生成する。訳文としての容認可能性は、日本語依存文法に基づいて検査する。すなわち、依存文法に基づく制約（特に、後方修飾性制約）を満たし、かつ、英文の語順に最も近い日本語文を同時通訳における訳文とする。ただし、日本語文として不自然でない場合に限り、倒置、すなわち、後方修飾性を満たさない依存関係を含む訳文についても生成を許容する。

本手法に基づき、英日同時通訳の実験システムを作成した。システムは、解析、変換、生成から構成され、英語対話文を形成するチャンクが先頭から入力されるたびに各処理を実行し、入力と同時進行的な訳文生成を実現する。実験では、ATIS コーパス [11] を使用し、翻訳品質、及び、同時性の観点から本手法を考察した。実験の結果、容認可能な程度の翻訳品質を保ち、かつ、原文の入力に対して高い同時性を備えた英日同時通訳における本手法の効果を確認した。

本論文の構成は以下のとおりである。2. では英日同時通訳者の訳出方略について論じる。次に 3. で、提案する訳文生成手法について述べ、4. で本手法を導入した同時通訳システムについて述べる。5. では、本システムによる英日同時通訳実験の結果について述べる。

## 2. 英日同時通訳者の訳出

本章では、英日同時通訳において英語と日本語の語順の違いにより生じる問題について述べる。

### 2.1 英日間の語順の違いによる問題

名古屋大学同時通訳データベース [12] は、同時通訳者による発話を大規模に収集した音声コーパスである。このデータベースに収録された旅行対話音声のうち、英語話者によって発話された対話文

(E1) You can pick your ticket up at the counter today.

(E2) There are three Sushi restaurants near this hotel.

の翻訳について考える。英語文 (E1)、(E2) の標準的な日本語訳は、例えば

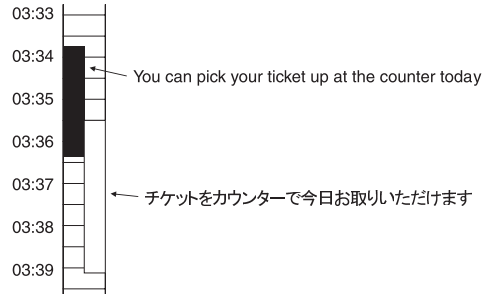


図 1 英語文 (E1) に対する同時通訳者の訳出例

Fig. 1 Simultaneous interpreter's translation for English sentence (E1).

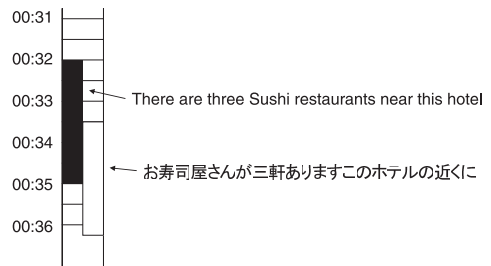


図 2 英語文 (E2) に対する同時通訳者の訳出例

Fig. 2 Simultaneous interpreter's translation for English sentence (E2).

(J1) 今日、カウンターでチケットをお取りいただけます。

(J2) ホテルの近くにお寿司屋が三軒あります。

である。日本語訳 (J1)、(J2) の訳出をできる限り早い段階で行ったとしても、文頭の「今日」「ホテル」に対応する英語 “today”, “hotel” が原文の文末に生起するため、話者の発話が終了する前に通訳者が発話を開始することは不可能である。

### 2.2 同時通訳者の訳出方法

一方、同時通訳データベースに収録されている、英語文 (E1) と (E2) に対する訳出の実例をそれぞれ図 1 と図 2 に示す。これらの図では、英語話者と英日同時通訳者の発話のタイミングを視覚化し表示している。図 1 と図 2 は、同時通訳者が話者の発話途中の段階で訳出を開始していることを示している。

ここで同時通訳者が訳出した日本語文を見てみると、同時通訳者は英語文 (E1) と (E2) を

(S1) チケットをカウンターで今日お取りいただけます。

(S2) お寿司屋さんが三軒あります、ホテルの近くに。

と訳出しており、これらは、標準的な日本語訳 (J1)、

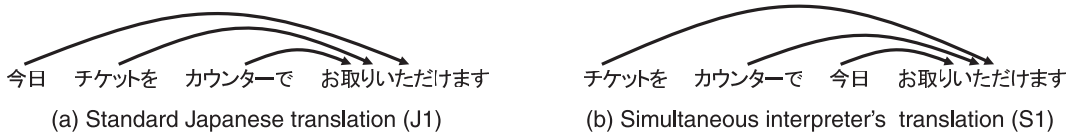


図 3 標準的な日本語訳 (J1) と同時通訳者による日本語訳 (S1) の依存構造  
 Fig. 3 Dependency structures of standard Japanese translation (J1) and simultaneous interpreter's translation (S1).

(J2) とは語順が異なる。具体的には、日本語訳「今日」と「ホテルの近くに」が、標準的な日本語訳では文頭に生起しているのに対して、同時通訳者の日本語訳では、文のより後方で生起している。これは、同時通訳者の日本語訳が、標準的な日本語訳に比べて、より英語文の語順に近くなる場合があることを意味している。このように、英日同時通訳者は、語順の入換えを巧みに活用し、日本語訳を英語文の語順に近い語順で訳出することにより、話者発話との同時進行的な訳出を可能にしている。

本論文では、このような同時通訳者の訳出方略を、計算機による英日同時通訳に応用することを試みる。

### 3. 訳文生成手法

本章では、2. で示した訳出方略を実現するための日本語訳文生成手法について述べる。本手法では、英語入力文を構成するチャンクの対訳表現（以下、対訳チャンク）が、その生起順序に従って順次、生成部に入力されることを前提とする。また、対訳チャンク間に成立する係り受け関係についても同様に順次入力されることとする。例えば、英語文 (E1) において “You can pick your ticket up” までが入力された段階では、「チケットを」「お取りいただけます」という対訳チャンクとその間の係り受け関係が生成部に入力されている。

本手法における中心的なアイデアは、訳文として容認可能な品質を保つという制約のもとで、できる限り早く対訳チャンクを出力するための制御方法にあり、容認可能性は依存文法に基づいて判定する。なお、本手法に基づく英日同時通訳システムについては 4. で詳述する。

#### 3.1 容認可能な日本語文とその依存構造

2. で述べた同時通訳者の日本語訳は、標準的な語順とは異なる語順で構成されているものの、対話翻訳システムが出力する日本語としては容認できる。本節では、そのような語順の入換えができる条件を依存文法

に基づいて検討する。

##### 3.1.1 後方修飾性

日本語には、文節間に係り受けと呼ばれる依存関係が存在し、係り文節が受け文節に依存するという関係として定めることができる。英語文 (E1) の標準的な日本語訳 (J1) と同時通訳者の日本語訳 (S1) の依存構造をそれぞれ図 3 の (a) と (b) に示す。図 3 の各矢印が係り受け関係を示しており、矢印の始点と終点がそれぞれ係り文節と受け文節を示している。標準的な日本語訳 (J1) の係り受け関係は、すべて前方の文節から後方の文節に係っている。このような性質は一般に後方修飾性と呼ばれる。一方、同時通訳者の日本語訳 (S1) も、標準的な日本語訳 (J1) に対して語順が入れ替わっているものの、すべての係り受け関係が後方修飾性を満たしている。係り受け関係における後方修飾性は、日本語における基本的な性質であり、係り受け関係が後方修飾性を満たしていることが、日本語として容認されるための重要な要因になると考えられる。

そこで本手法では、システムが生成する日本語訳文がたとえ標準的な語順でなくても、すべての係り受け関係が後方修飾性を満たしていれば訳文として容認するというように、訳文の語順に関する制約を緩和し、より原文の語順に近い語順の訳文を生成する。

##### 3.1.2 倒置

英語と日本語とでは、述部の生起する位置が大きく異なる。一般に、英語では述部が文頭に近い位置に生起するのに対して、日本語では文末に生起する。そのため、述部は早い段階で入力されてもその訳出は最後となり、述部の訳出遅延が増大するという問題が発生する。この問題は、前項で述べた語順の制約の緩和だけでは解決できない。例えば、図 4 (a) の日本語訳 (J2) の依存構造では、述部「あります」は、他のすべての対訳チャンク「近くに」、「お寿司屋さん」、「三軒」の係り先であるため、後方修飾性を満たすには文末で述部を生成する以外にない。

一方、同時通訳者の訳出例 (S2) の依存構造

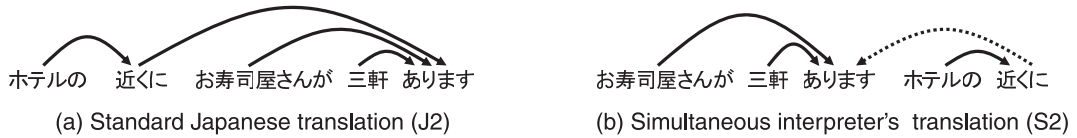


図 4 標準的な日本語訳 (J2) と同時通訳者による日本語訳 (S2) の依存構造  
 Fig. 4 Dependency structures of standard Japanese translation (J2) and simultaneous interpreter's translation (S2).

表 1 係り文節の数と倒置の起こる割合  
 Table 1 Number of dependent bunsetsus and rate of inversion.

文節数	全体	倒置	割合 (%)
1	2023	32	1.58
2	1376	103	7.49
3 以上	600	95	15.83

(図 4(b)) では、述部「あります」を文末ではなく、「ホテルの近くに」より前方で訳出している．そして、「近くに」から「あります」に係る係り受け関係が後方修飾性を逸脱している．本論文の以下では、後方修飾性を逸脱する係り受け関係を倒置と呼ぶ．倒置は話し言葉においてたびたび起こるため、倒置を含む日本語文であっても容認できる可能性がある．倒置を許容することにより、より原文の語順に近い訳文を生成できることが期待される．

しかし、その一方で、倒置がむやみに現れると聞き手の理解に支障をきたす可能性がある．倒置が許容される条件を明らかにするために、日本語話し言葉における倒置の特徴を調査した．調査には、名古屋大学 CIAIR 音声対話コーパス [13] における係り受け関係が付与された日本語発話 7781 ターンを利用した．その中には、倒置が 230 個存在した．調査結果より、倒置には、

1. 受け文節の 92.2% が述部である
2. 受け文節に係る係り受けの数が多いほど倒置が起りやすい (表 1 参照)
3. 1 文内に生起する倒置は 2 個以内であるという特徴があることが分かった．

本研究では、これらの特徴を満たす倒置の生成を許容する．すなわち、生成する日本語訳がすべての係り受け関係が後方修飾性を満たすか、あるいは、倒置となる係り受けが、

1. 倒置の受け文節が述部である
2. 倒置の受け文節に係る文節の数が  $L$  以上である ( $L$  の値は実験により決定)
3. 1 文内に生起する倒置は 2 個以内である

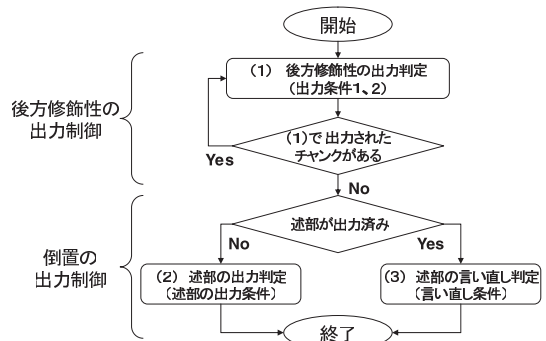


図 5 本手法のアルゴリズム  
 Fig. 5 Algorithm of our method.

を満たすこととする．これらの条件を設けることにより、訳質の低下を抑えつつも語順の制約を緩和することができる．

### 3.2 アルゴリズム

本節では、前節で示した語順制約の緩和を活用して、より原文の語順に近い訳文を入力と同時的に生成する方法について述べる．本手法では、入力との同時的な生成のために、新しい対訳チャンクが生成部へ入力されるたびに、図 5 に示すアルゴリズムを実行する．アルゴリズムは、後方修飾性を満たすための出力制御と、倒置を活用するための出力制御の 2 段階で構成される．

英文 “I’ll go to the airport with my friends by taxi next Monday.” に対応する対訳チャンク「行きます」、「空港へ」、「友達と」、「タクシーで」、「来週の月曜日に」が生成部にこの順番で入力された場合の処理の例を図 6 に示す．図 6 の各列は、左から順に「生成部に入力された対訳チャンク」、「依存構造」、「各処理時点を表す記号」、「実行される出力判定 (図 5 の (1) (2) または (3) のいずれか)、及び、判定される対訳チャンク」、「出力」に対応している．これ以降では、図 6 の訳出例を用いて、処理の詳細について説明する．

#### 3.2.1 後方修飾性の出力制御

もし、すべての対訳チャンクが生成部に入力されて

Input	Dependency structure	Symbol	Processing of output judgment	Output
行きます	行きます	(a1)	(1) 「行きます」	
		(a2)	(2) 「行きます」	
空港へ	行きます 空港へ	(b1)	(1) 「行きます」, 「空港へ」	
		(b2)	(2) 「行きます」	
友達と	行きます 空港へ 友達と	(c1)	(1) 「行きます」, 「空港へ」, 「友達と」	空港へ
		(c2)	(1) 「行きます」, 「友達と」	空港へ
		(c3)	(2) 「行きます」	空港へ 行きます
タクシーで	行きます 空港へ 友達と タクシーで	(d1)	(1) 「友達と」, 「タクシーで」	空港へ 行きます 友達と
		(d2)	(1) 「タクシーで」	空港へ 行きます 友達と
		(d3)	(3) 「行きます」	空港へ 行きます 友達と
来週の月曜日に	行きます 空港へ 友達と タクシーで 来週の月曜日に	(e1)	(1) 「タクシーで」, 「来週の月曜日に」	空港へ 行きます 友達と タクシーで
		(e2)	(1) 「来週の月曜日に」	空港へ 行きます 友達と タクシーで
		(e3)	(3) 「行きます」	空港へ 行きます 友達と タクシーで
S	行きます 空港へ 友達と タクシーで 来週の月曜日に S	(f1)	(1) 「来週の月曜日に」	空港へ 行きます 友達と タクシーで 来週の月曜日に
		(f2)	(3) 「行きます」	空港へ 行きます 友達と タクシーで 来週の月曜日に 行きます

図 6 出力処理の例

Fig. 6 Example of output processing.

から訳出するのであれば、すべての係り受け関係が後方修飾性を満たす訳文を生成するのは難しいことではない。すなわち、

[出力条件 1] 自らを係り先とするすべての対訳チャンクが出力済みである

を満たした対訳チャンクを出現順に出力すればよい。しかし、入力と同時に訳出する場合には、たとえ入力途中で出力条件 1 を満たした対訳チャンクを出力しても、その後新たに入力された対訳チャンクがその出力済みの対訳チャンクに係る場合には、後方修飾性を逸脱することになる。例えば、図 6(a1) の時点では、「行きます」だけが入力されているので、この時点では出力条件 1 を満たすためそのまま出力できる。しかし、次に入力される「空港へ」が「行きます」に係るため後方修飾性を逸脱する。

そこで、後から入力される対訳チャンクの係り先となる可能性を考慮するために、出力条件 1 に加えて、[出力条件 2] 新たに生成部に入力された対訳チャンクではない

を満たすこととする。例えば、図 6(a1) の時点で、「行きます」は自らに係る対訳チャンクがないため出力条件 1 を満たすものの、入力されたばかりの対訳チャンクであり、出力条件 2 を満たさないため、出力しない。もちろん、出力条件 2 を用いても後方修飾性を逸脱することはあるが、新しく入力された対訳チャンクがその直前の対訳チャンクに係るかどうかを検出できるので、後方修飾性を逸脱する可能性は低くなる。

出力制御のアルゴリズムでは、出力条件 1, 2 を用いた出力判定 (1) の後に、後方修飾性の判定処理で出力されたチャンクがある場合には、再度、後方修飾性

の出力判定 (1) を実行している (図 5 参照)。これは、対訳チャンクが出力されることにより、他の対訳チャンクが新たに出力条件 1 を満たす場合があるためである。

### 3.2.2 倒置の出力制御

倒置の出力制御では、3.1.2 で述べた倒置が許容される条件の 1. ~ 3. を満たす訳文を生成する。倒置の条件 1 より、倒置の受け文節は述部に限られるため、述部に相当する対訳チャンクに対する出力条件を緩和する。また、倒置の条件 2 より、述部を出力していない場合には、

[述部の出力条件] 自らを係り先とする対訳チャンクのうち  $L$  個以上が出力済みである

を述部が満たせば、それを出力する。図 6 は、述部の出力条件が  $L = 1$  であった場合の処理の例である。「友達と」まで入力された時点 (図 6(c1) 参照) では、出力条件 1 と 2 の双方を満たすので、「空港へ」を出力する。次に、「友達と」は出力条件 2 を満たさないため、出力しない。そして、述部「行きます」は、出力条件 1 を満たさないため後方修飾性の出力判定 (1) では、出力しない。しかし、述部の出力判定 (2) では、出力済みである「空港に」が係っているので、述部に係る文節が 1 個以上出力済みとなり述部の出力条件を満たすため、「行きます」を出力する (図 6(c2), (c3) 参照)。

しかし、述部の出力条件のみによる制御では、述部を出力した後にその述部に係る対訳チャンクが三つ以上出力される可能性があり、その場合に倒置の条件 3. を逸脱する。例えば、図 6(f1) では、「友達と」、「タクシーで」、「来週の月曜日に」から述部「行きます」への係り受け関係が倒置になっている。そこで、述部が出



力済みであるときに、

[言い直し条件] 倒置が3個以上発生しているを満たす場合には、述部をもう一度出力することにより倒置の生成を回避する。図6 (f2) では、倒置となる係り受けが3個に達するため言い直し条件を満たす。述部「行きます」を言い直した結果、倒置となる係り受けの生成が解消されている。

### 3.2.3 複雑な文の出力処理の例

上述の例では、すべての対訳チャンクが主節の述部に係る単純な文を対象に説明を与えたが、より複雑な文でも同様に処理することができる。

例えば、前置詞句を伴う長い名詞句を含む文の場合でも、対訳チャンク間の修飾関係に基づき処理を実行する。“I need a flight from Minneapolis to Long Beach on June twenty six.” では、三つの前置詞句がいずれも“a flight”に係るため、名詞句の訳出は、対訳チャンク「ミネアポリスから」、「ロングビーチへ」、「6月26日の」をこの順に出力してから「フライトを」を出力する。

また、連体修飾節や副詞節などの従属節を含む複文においても、同様に、出力条件に従って出力を制御する。例えば、“Show me the flights that go from Baltimore to Seattle.” では、“that”以下は“the flights”に依存する連体修飾節であり、“that”以降の出力後に“the flights”の対訳チャンクを出力することになる。すなわち、「私にボルティモアからシアトルへ行く飛行機を示して下さい」と訳出する。また、“You must contact us if you want to cancel your reservation.” においては、まず、“us”の対訳チャンク「私々に」を訳出する。ここで、述部の出力条件 ( $L = 1$  の場合) が満たされるため、この時点で、“must contact”の対訳チャンクを訳出する。最後に、“if”以降の副詞節の対訳チャンクを訳出する。「私々に連絡しなければなりません、予約をキャンセルしたいならば」のように、倒置を含む文として出力される。

## 4. システム構成

3. で示した訳文生成手法を実装した英日同時通訳システムの構成を図7に示す。

本システムは、構文トランスファー方式に基づいており、英語の依存構造から日本語の依存構造への変換により訳文を漸進的に作り上げる。依存構造の変換として実現することにより依存文法に基づく検査を容易に実行することができ、3.2 で示した出力制御アルゴ

リズムを自然に導入することができる。なお、同時翻訳に適した対訳が与えられたデータを大量に利用することができれば、本手法をコーパスを用いた方式によって実現できる可能性がある。

本システムは、解析部、変換部、生成部から構成されている。解析部では、入力された英単語列の依存構造を漸進的に解析する。変換部では、英単語列の依存構造に対して構造変換及び語彙変換を実行し、日本語の依存構造に変換する。生成部は、日本語の依存構造をもとに3.2で示した出力制御アルゴリズムを用いて日本語訳を生成する。述部の出力条件が  $L = 2$  の場合に、英文

(E3) I want to fly from San Francisco to Denver next Monday.

の“Denver”が入力されたときの処理の流れを図8に示す。以下では図8を用いて各部の処理を説明する。

### 4.1 解析部

解析部では、入力された単語列を辞書を用いて品詞タグ付けし、単語とその品詞情報を用いて baseNP [14] をまとめあげる (図8の(1)参照)。次に、文脈自由文法に基づく漸進的依存構造解析手法 [15] に従って入力された英文の依存構造を漸進的に解析する。この解析手法では、主辞付き構文規則を用いて漸進的構文解析法に従い入力文の断片に対する句構造を生成する (図8の(2)参照)。また、図8の(2)の NP(?) のようにまだ入力されていない語 (未入力語) についても

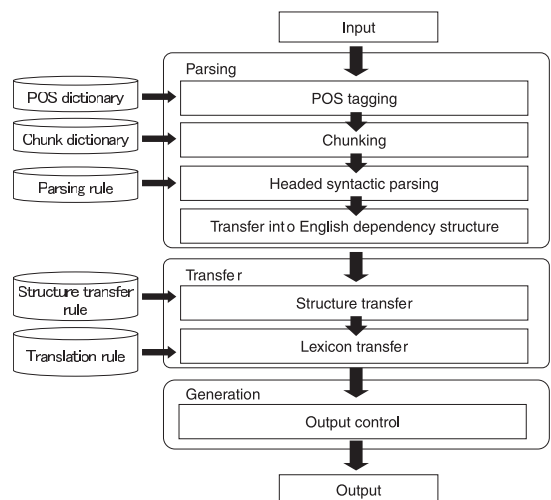


図7 システムの構成図

Fig. 7 Configuration of our system.

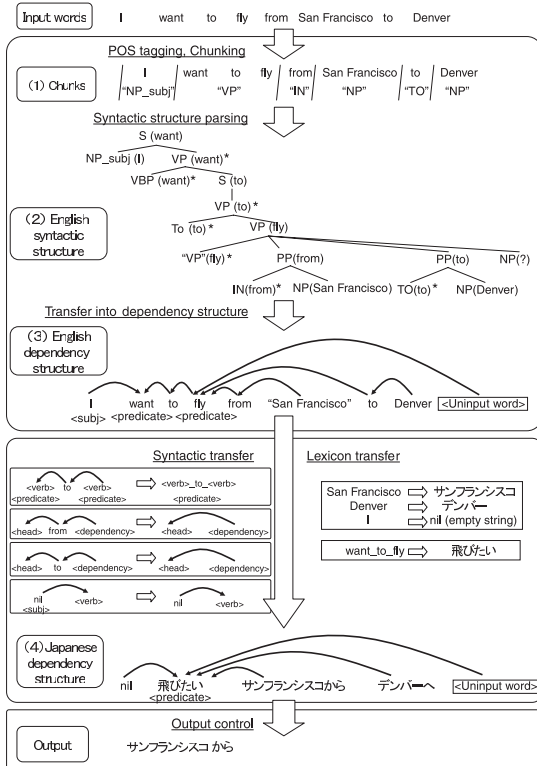


図 8 英文 (E3) の “Denver” が入力されたときの処理の流れ (ただし、述部の出力条件のしきい値  $L = 2$ )  
 Fig. 8 Translation flow for fragment “I want to fly from San Francisco to Denver” (in  $L = 2$ ).

予測し解析する．続いて、句構造の主辞情報の品詞をもとに、Collins の規則 [16] に従って句構造をチャンク間の依存構造に変換する (図 8 の (3) 参照)．

一般に、漸進的に構文解析すると、入力文の断片に対して膨大な数の構文構造が生成されるという曖昧性の問題が発生する．本システムでは、構文解析に用いる文法規則に確率を付与し、構文構造の生成に利用した規則の確率からその構文構造の確率を求め、確率が最大となる構文構造を選択する．

文法規則には、主語 (subject) と目的語 (object) の情報が付与されており、構文解析することにより主語と目的語になるチャンクを解析する．更に、品詞が動詞である単語を述部 (predicate) とみなす．図 8 (3) の例では、主語は “I” であり、述部は “want” と “fly” の二つである．

#### 4.2 変換部

変換部では、構造変換と語彙変換を実行することにより、英語の依存構造を日本語の依存構造に変換する

(図 8 の (4) 参照)．構造変換では、構文変換規則により、助動詞や to 不定詞を本動詞と結び付け、述部をまとめたり、前置詞をそれに係るチャンクの助詞に変換するなどの処理を行う．例えば、“want to fly” は述部として一つのチャンクにまとめる．前置詞 “from” は、それに係る “San Francisco” に付属する助詞 “から” に変換する．語彙変換では、語彙変換規則を用いて対訳チャンクに変換する．

#### 4.3 生成部

生成部では、3.2 で説明した出力制御アルゴリズムに従い訳出する．図 8 の例では、「サンフランシスコから」は、それに係る対訳チャンクがないため、出力する．一方、「デンバーへ」は、新たに入力された対訳チャンクであるため、この段階では出力しない．また、述部「飛びたい」には、未出力の「デンバー」が係り、かつ、「飛びたい」に係る訳出済みの対訳チャンクは「サンフランシスコから」ただ一つであるため、述部の出力条件 ( $L = 2$ ) を満たさないことから、この段階では出力しない．

#### 4.4 同時翻訳処理の流れ

本システムにおいて、英文 (E3) が入力されたときの入力途中の構文構造、英語の依存構造、日本語の依存構造、及び、出力を図 9 に示す．図 9 の日本語の依存構造の日本語訳における下線は、その時点で出力可能と判定された日本語訳を示している．

### 5. 同時翻訳システムによる翻訳実験と評価

本手法により、日本語として容認可能で、かつ、話者の発話に対して同時的に訳文を生成できることを示すために、本手法を実装した同時翻訳システムを用いて翻訳実験を行った．実験では、翻訳品質と同時性の観点から本手法を評価した．

#### 5.1 翻訳品質と同時性の評価指標

翻訳品質と同時性を評価する指標について説明する．翻訳品質の評価には、訳文を 4 段階で評価した結果を用いた．評価基準を表 2 に示す．

一方、同時性の評価尺度としては、訳出遅延チャンク数を用いる．訳出遅延チャンク数とは、あるチャンクが入力されてからチャンクに対応する訳が出力されるまで (ただし、言い直しされた場合は、言い直し表現が出力される時点まで) に入力されたチャンクの数である．以下の式で、平均訳出遅延チャンク数  $D$  を求める．

Input	Syntactic structure	English dependency structure	Output
		Japanese dependency structure	
I			
want to fly			nil
from			
San Francisco			
to			サンフランシスコから
Denver			
next Monday			デンバーへ 飛びたい
\$			来週の月曜日に

図 9 英文 (E3) の処理の流れ

Fig. 9 Processing flow for English sentence “I want to fly from San Francisco to Denver next Monday.”



表 2 評価基準  
Table 2 Evaluation standard.

評価ランク	説明
A (Perfect)	会話文として問題がない
B (Fair)	会話文としてごちない、助詞や言葉遣いに間違いがあるものの、言いたいことはほとんど分かる
C (Acceptable)	訳文は情報が欠けていたり、断片的な情報しか分からないが、原文の主要な情報は正しく理解できる
D (Nonsense)	会話文として重要な情報が抜けているか、原文の意味が正しく理解できない

$$D = \frac{\sum_{k \in K} d_k}{n} \quad (1)$$

ここで  $d_k$  は、あるチャンク  $k$  の訳出遅延チャンク数を、 $n$  は全入力文の総チャンク数を、 $K$  は全入力文の全チャンクの集合を表す。

訳文の同時性を評価する別の指標として話者・通訳者発話時間を計測する。話者・通訳者発話時間とは、原文の入力を開始してから訳出が終わるまでの時間である。本研究では、入力と出力は、ともに一定の話速 (7.43 mora/s) で行われると仮定して<sup>(注1)</sup>、算出した。

### 5.2 実験環境と実験データ

本システムを Windows PC (Intel Core2 CPU 1.86 GHz, 2 GByte RAM) 上に JRE1.6.0 を用いて実装した。翻訳対象としては、構文木が付与された英語対話文である必要があり、本実験では、Penn Treebank [11] に収録されている ATIS コーパスの全 578 文を用いた。578 文の長さの分布を図 10 に示す。1 文当たりの平均単語数 (チャンク数) は、7.53 単語 (5.66 チャンク) であり、最大単語数 (チャンク数) は 34 単語 (29 チャンク) である。比較的長い文 (11 単語以上の文) も 103 文含まれており、翻訳処理における同時性の効果の検証に利用できる。578 文の種類の内訳は、単文が 297 文、述部を含まない名詞句や前置詞句が 220 文、関係代名詞節や連体修飾節を含む文が 39 文、そのほかに重文などが 21 文であった。

構文解析に用いた文法規則として、ATIS コーパスに付与された構文木データから獲得した 327 規則を用いた<sup>(注2)</sup>。翻訳対象の文に対して翻訳家が参照訳を作成し、その参照訳をもとにチャンク単位で対訳辞書を作成した。原文を漸進的に構文解析する処理では、解析途中の構文候補が膨大な数となるため、解析効率の観点から、解析途中の構文候補に付与された確率が  $1.0 \times 10^{-15}$  以下になる場合には、その構文候補を枝刈りした。

### 5.3 実験結果

#### 5.3.1 翻訳品質

本手法の有効性を示すために、実験システムの翻訳

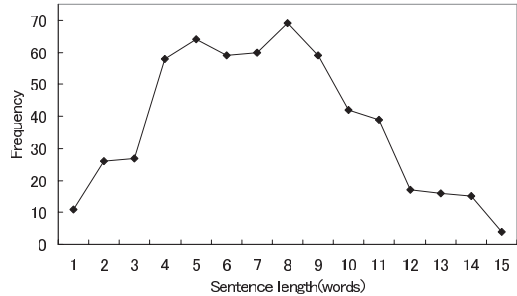


図 10 文の長さの分布  
Fig. 10 Distribution of sentence length.

品質を評価した。倒置の活用が品質に与える影響を調べるために、後方修飾性の出力制御のみを利用して訳出する場合 (すなわち、倒置を活用しない場合) と、後方修飾性に加えて倒置を活用 ( $L = 2$ ) して訳出する場合の 2 種類で実験した。また、本手法の比較対象として、原文の語順どおりに訳出する方法 (訳出の同時性を重視した方式)、並びに、原文がすべて入力されてから翻訳を開始する方法 (以下、文単位の手法) (翻訳品質を重視した方式) を実装し、それぞれ評価した。なお、文単位の手法では、対訳辞書と文法規則は本手法と同じものを使用し、また、標準的な語順で生成するための生成規則を新たに作成し使用した。

実験結果を表 3 に示す。容認可能性の観点から、翻訳品質が A, B, C のいずれかである場合を正解とした。本手法 (後方修飾性の出力制御のみを利用した場合) の翻訳精度 (すなわち、A + B + C の割合) は 71.6% であった。原文の語順どおりに訳出した場合と比べて 25.4% 高く、また、文単位の手法に対しては 4.4% の低下にとどまっており、容認可能な程度の翻訳品質が保たれていることが分かった。

一方、本手法のうち、倒置導入による翻訳品質への影響を調べるため、「倒置なし」と「倒置あり」とで結果を比較すると、倒置が発生することによって新たに

(注1): 名古屋大学同時通訳データベースの対話データを用いて話速を調査し、その結果に基づいて設定した。

(注2): 文献 [17] の手法に従って文法規則を変換して、獲得した。

表 3 翻訳品質の実験結果  
Table 3 Translation quality.

手法	翻訳品質			
	A	A + B	A + B + C	D
本手法 (後方修飾性のみ)	116 (20.1%)	351 (60.7%)	414 (71.6%)	164 (28.4%)
本手法 (後方修飾性 + 倒置 $L = 2$ )	111 (19.2%)	312 (54.0%)	411 (71.1%)	167 (28.9%)
原文の語順どおり	15 (2.6%)	61 (10.6%)	267 (46.2%)	311 (53.8%)
文単位的手法	121 (20.9%)	372 (64.4%)	439 (76.0%)	139 (24.0%)

翻訳誤りとなった文は3文であった。翻訳精度の低下は0.5%であり、倒置利用の許容可能性が示された。ただし、高い翻訳品質を備えるA+Bに着目すると、その割合は、倒置を許容することによって、6.7%低下することとなった。これは、「倒置なし」においてAまたはBと評価された文のうち、倒置が発生したものが40文存在し、そのうち39文において翻訳品質がCに低下したためである。例えば、英語文 “On Tuesday I'd like to fly from Detroit to Saint Petersburg.” は、「倒置なし」では「火曜日にデトロイトからセントペテルスブルクへ乗りたい」と訳出され翻訳品質がAと判定されたが、「倒置あり」では「火曜日にデトロイトから乗りたいセントペテルスブルクへ」という訳出になり翻訳品質がCと判定された。このように、訳文に倒置が含まれると、きれいな訳文 (fair) であるとは評価されず、倒置を許容する限り品質の低下を避けることは難しい。なお、品質が低下しなかった1文は、構造上は後方修飾に逸脱する依存関係が存在するものの、表層上では倒置が生じているようには見えない訳文が偶然生成されたことによる。

本手法は、文が長くなるほど同時性においてその効果が大きくなると考えられるが、その一方で、文が長くなり複雑さが増すと、翻訳品質が低下する可能性がある。そこで、入力文の長さ、並びに、複雑さと翻訳品質との関係を調べた。文の長さとの関係を図11に、文の複雑さとの関係を図12に、それぞれ翻訳方式 (倒置あり、倒置なし、文単位) ごとに示す。なお、文の複雑さとしては、文に含まれる動詞の数を用いた。図11において、本手法は文が長くなると翻訳精度が低下するものの、文単位的手法との間で大きな差はない。また、図12では、文の複雑さに対する低下傾向は、本手法と文単位的手法とでほぼ同じであり、同時性の効果が示される文においても品質はある程度保たれることが示された。

本手法 ( $L = 2$ ) における翻訳誤りの原因を表4に示す。なお、誤りの原因が複数ある場合には重複を許

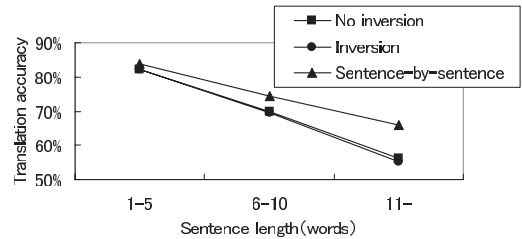


図 11 文の長さとの翻訳精度の関係

Fig. 11 Relation between sentence length and translation accuracy.

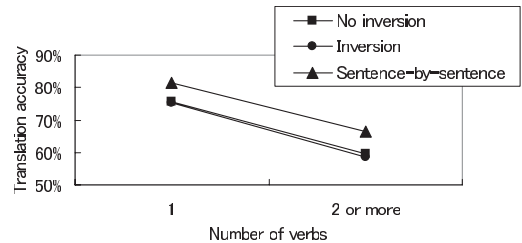


図 12 文の複雑さとの翻訳精度の関係

Fig. 12 Relation between number of verbs and translation accuracy.

表 4 翻訳誤りの原因  
Table 4 Causes of incorrect translations.

原因	文数
語順の問題	74
語彙のあいまい性	61
不適格性	57
その他	8

している。語順誤りのうち、等位接続詞に関するものが33文、Wh-疑問詞に関するものが36文存在した。

等位接続詞を含む文では、英語と日本語とでその依存構造が対応しない場合があることが主な要因である。本手法で用いた主辞付与規則 [16] に従えば、例えば “X and Y” の場合、先頭の “X” が主辞となり、“X” に “and” と “Y” が依存することになる。それに対して、日本語「X と Y」では、「X と」が「Y」に依存する。この問題を解決する方法として、等位接続詞を含む構文規則の主辞を後方の “Y” とするよう主辞付

表 5 同時性の評価  
Table 5 Evaluation of simultaneity.

手法	平均訳出遅延 チャンク数	話者・通訳者 発話時間 (秒)
本手法 (後方修飾性のみ)	2.13	5.78
本手法 (後方修飾性 + 倒置 $L = 2$ )	2.08	5.75
文単位的手法	3.61	6.88

表 6 述部の訳出制御の設定を  $L = 1, 2, 3$  にした場合の実験結果  
Table 6 Result in threshold value  $L = 1, 2, 3$ .

$L$ の設定値	翻訳精度			平均訳出遅延 チャンク数	倒置を 含む文数	言い直しを 含む文数
	A	A + B	A + B + C			
1	58 (10.0%)	200 (34.6%)	347 (60.0%)	1.87	270	15
2	111 (19.2%)	312 (54.0%)	411 (71.1%)	2.08	62	1
3	116 (20.1%)	345 (59.7%)	414 (71.6%)	2.12	18	0

与規則を修正することが考えられる。これを採用すると、33 文のうち 20 文 (60.6%) について問題の改善が見込まれる。特に、このような等位接続詞は長い文に含まれることが多く、その対処により長い文の翻訳精度が向上することが期待できる。

一方、Wh-疑問詞を含む文における語順誤りは、“Which is the latest flight?” (標準訳:「最終便のフライトはどちらですか」)のように、補語が主語よりも先に出現する文において生じるものであり、依存構造に従って訳出すると be 動詞と補語が分離することになるためである。この問題を解消するには、Wh-疑問詞と be 動詞をまとめる変換規則を追加すればよく、これにより、ほぼ 100%改善されると予想される。

なお、語彙的あいまい性や不適格性の問題は、同時翻訳に特有のものではなく、すべて一般的な対話翻訳においても同様に生じる原因である。

### 5.3.2 同時性

入力との同時性を評価するために、平均訳出遅延チャンク数と平均話者・通訳者発話時間を計測した。本手法の後方修飾性の出力制御のみを利用して訳出した場合、本手法の後方修飾性の出力制御と倒置を活用 ( $L = 2$ ) して訳出した場合、文単位的手法で訳出した場合の平均遅延チャンク数と平均話者・通訳者発話時間を表 5 に示す。文単位的手法に比べて、本手法では平均訳出遅延チャンク数、話者・通訳者発話時間がともに減少しており、入力との同時性が向上したことが確認できた。また、倒置を活用する場合の方が倒置を活用しない場合に比べて、平均訳出遅延チャンク数がわずかであるが減少しており、倒置の活用が同時性の向上に有効であることが確認できた。

本手法と文単位の手法の場合の話者発話時間と話者

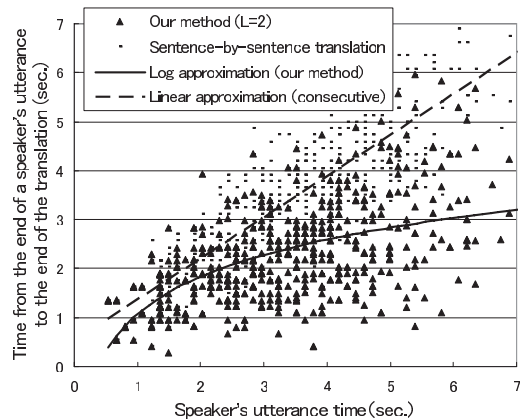


図 13 話者発話時間と話者発話終了から訳出終了までの時間の関係

Fig. 13 Relation between the speaker's utterance time and the time from the end of a speaker's utterance to the end of the translation.

発話終了から訳出終了までの時間の関係を図 13 に示す。図 13 では、本手法と文単位の手法の回帰方程式を与えた<sup>(注3)</sup>。図 13 より、文単位の手法に比べ本手法を用いることにより、話者発話時間が長くなるほど訳出終了までの時間が減少しており、本手法は長い発話であるほど同時性の向上が期待できることを確認した。

### 5.3.3 倒置の活用

述部の出力条件の  $L$  の値を 1 から 3 まで変更した場合の実験結果を表 6 に示す。 $L = 1$  と  $L = 2$  の場合を比較すると、 $L = 1$  では 270 文で倒置が活用されて

(注3)：本手法の回帰方程式は対数回帰で、文単位の手法は線形回帰で回帰法式的を求めている。これは、線形回帰と対数回帰で回帰方程式を計算したときに、決定係数が大きくなる回帰方程式をそれぞれ採用した。決定係数が大きいほど、実際のデータが回帰直線の周辺に分布していることを意味している。

おり、同時性は向上するものの、翻訳精度は 11.1% 低下した。  $L = 2$  と  $L = 3$  の場合を比較すると、  $L = 2$  の方がわずかに同時性が向上する一方で、翻訳精度の低下はほとんど見られなかった。

## 6. む す び

本論文では、日本語訳の構成要素間の依存構造に基づいて訳文を生成する英日同時翻訳手法を提案した。本手法を導入した英日同時翻訳システムを用いて、ATIS コーパス 578 文の翻訳実験を行った。その結果、本手法による翻訳精度は 71.6% であり、翻訳品質において、本手法の利用可能性を確認した。また、文単位の手法と比較して、本手法の方が遅延チャンク数と話者・通訳発話時間が減少しており、同時性が向上することを示した。

本論文では、日本語訳文の生成において、語順の入換えを積極的に行うことにより翻訳処理の高い同時性を実現する手法を提案した。しかし、すべての対話文に対して本手法を適用すると、結果的に倒置や言い直しが多頻度で現れることになり、翻訳品質の観点からは必ずしも好ましくない。したがって、本手法を状況に応じて選択的に利用することが考えられる。例えば、長い文や話速が速い文を翻訳する場合、そのまま逐次的に訳出すると遅延が大きくなる傾向にあるため、本手法を適用することが有効であると予想される。対話文の性質と手法の効果の程度との関係について、今後、考察していく予定である。

謝辞 御指導下さる名古屋大学教授の坂部俊樹先生に深く感謝致します。また、本研究にあたり有益な貢献をして頂きました元名古屋大学大学院の渡邊善之氏、水野敦氏、訳文の品質評価に御協力頂いた翻訳家の和氣祥子氏に感謝致します。本研究は、一部、科学研究費補助金（基盤研究（B））（No.20300058）により実施したものである。

## 文 献

- [1] R. Frederking, A. Blackk, R. Brow, J. Moody, and E. Stein-brecher, "Field testing the tongues speech-to-speech machine translation system," Proc. 3rd International Conf. on Language Resources and Evaluation, pp.160-164, May 2002.
- [2] R. Isotani, K. Yamada, S. Ando, K. Hanazawa, S. Ishikawa, and K. Iso, "Speech-to-speech translation software on PDAs for travel conversation," NEC Res. and Dev., vol.42, no.2, pp.197-202, April 2003.
- [3] F. Liu, Y. Gao, L. Gu, and M. Picheny, "Noise robustness in speech to speech translation," IBM Tech. Report, RC22874, May 2003.
- [4] S. Nakamura, K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto, "The ATR multilingual speech-to-speech translation," IEEE Trans. Audio, Speech and Language Processing, vol.14, no.2, pp.365-376, March 2006.
- [5] 松原茂樹, "同時通訳の工学と科学—次世代自動通訳技術の実現に向けて," 情報処理, vol.49, no.6, pp.617-623, June 2008.
- [6] 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—逐次通訳との比較に基づく対訳コーパスの分析," 通訳研究, no.3, pp.32-52, Dec. 2003.
- [7] 萩原直之, "英日同時通訳作業に必須とされる time lag の翻訳の合理化による短縮とその裨益," 英米評論, 桃山学院大学紀要, no.5, pp.25-74, Dec. 1991.
- [8] J.W. Amtrup, "Chart-based incremental transfer in machine translation," Proc. 6th International Conference of Theoretical and Methodological Issues in Machine Translation, pp.188-195, July 1995.
- [9] F. Casacuberta, E. Vidal, and J.M. Vilar, "Architectures for speech-to-speech," Proc. Workshop on Speech-to-Speech Translation: Algorithms and System, pp.39-44, July 2002.
- [10] H. Mima and O. Furuse, "Simultaneous interpretation utilizing example-based incremental transfer," Proc. 17th International Conference on COLING and 36th Conference of the ACL, pp.855-861, Aug. 1998.
- [11] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," Computational Linguistics, vol.19, no.2, pp.310-330, April 1993.
- [12] H. Tohyama, S. Matsubara, N. Kawaguchi, and Y. Inagaki, "Construction and utilization of bilingual speech corpus for simultaneous machine interpretation research," Proc. 9th European Conf. on Speech Communication and Technology, pp.1585-1588, Sept. 2005.
- [13] S. Matsubara, T. Murase, N. Kawaguchi, and Y. Inagaki, "Stochastic dependency parsing of spontaneous Japanese spoken language," Proc. 19th International Conf. on Computational Linguistics, vol.1, pp.640-645, Sept. 2002.
- [14] K.W. Church, "A stochastic parts program and noun phrase parser for unrestricted text," Proc. 1st Conf. on Applied Natural Language Processing, pp.136-143, Feb. 1988.
- [15] 加藤芳秀, 松原茂樹, 外山勝彦, 稲垣康善, "主辞情報付き文脈自由文法に基づく漸進的な依存構造解析," 信学論 (D-II), vol.J86-D-II, no.1, pp.86-97, Jan. 2003.
- [16] M.J. Collins, "A new statistical parser based on bigram lexical dependencies," Proc. 34th Annual Meeting of the Association for Computational Linguistics, pp.184-191, June 1996.

- [17] B. Roark and M. Johnson, "Efficient probabilistic top-down and left-corner parsing," Proc. 37th Annual Meeting of the Association for Computational Linguistics, pp.421–428, June 1999.

(平成 20 年 9 月 16 日受付, 21 年 1 月 8 日再受付)



筧 浩一朗 (正員)

2002 名大・工・情報卒. 2004 同大大学院情報科学研究科博士前期課程了. 2007 同大学院後期課程了. 同年, 同大研究員を経て, 2009 名古屋大学大学院国際開発研究科助教. 自然言語処理, 音声言語処理の研究に従事. 情報処理学会, 言語処理学会

各会員.



松原 茂樹 (正員)

1993 名工大・工・電気情報卒. 1998 名大大学院博士課程了. 博士(工学). 同年, 同大助手を経て, 2002 名古屋大学助教授. 現在, 同情報基盤センター准教授. この間, 日本学術振興会特別研究員, ATR 音声言語コミュニケーション研究所客員研究員,

情報通信研究機構研究員を兼任. 自然言語処理, 情報検索, デジタル図書館の研究に従事. 情報処理学会, 人工知能学会, 言語処理学会, IEEE, ACM 各会員.



稲垣 康善 (名誉員)

1962 名大・工・電子卒. 1967 同大大学院博士課程了. 同大助教授, 三重大学教授を経て, 1981 名古屋大学工学部教授. 1997 同工学部長・工学研究科長, 2003 名古屋大学名誉教授, 愛知県立大学情報科学部教授,

2007 愛知県立大学名誉教授, 愛知工業大学経営情報科学部教授, 2008 豊橋技術科学大学理事・副学長. 工博. コンピューテーションとコミュニケーションの理論, オートマトン言語理論, ソフトウェア基礎論, 自然言語処理に関する研究に従事. 本会情報・システムソサイエティ会長, 副会長等を歴任, 功績賞受賞. 情報処理学会名誉会員・フェロー, 日本ソフトウェア科学会, 人工知能学会, 言語処理学会, IEEE, ACM, EATCS 各会員.