

# 生成方向を考慮した統計的機械翻訳のための デコーディングアルゴリズム

渡辺 太郎<sup>†</sup> 隅田 英一郎<sup>†</sup> 奥 乃 博<sup>††</sup>

統計的機械翻訳では、入力文に対して最適な翻訳を求める問題を探索問題としてとらえている。従来の探索システム、デコーダは入力文に対して文頭から出力文を生成していた (left-to-right デコーディングと呼ぶ)。本稿では、文末から出力を生成する right-to-left デコーディング、さらに、双方向から生成する双方向デコーディングを提案する。日本語、英語による翻訳実験の結果、日英翻訳においては left-to-right デコーディングが適しており、また、英日翻訳においては、right-to-left デコーディングが良いことが分かった。双方向デコーディングアルゴリズムではさらに探索エラーを減少させることが確かめられた。

## Decoding Algorithms for Statistical Machine Translation Considering Generation Directions

TARO WATANABE,<sup>†</sup> EIICHIRO SUMITA<sup>†</sup> and HIROSHI G. OKUNO<sup>††</sup>

The framework of statistical machine translation formulates the problem of finding the optimum translation as a search problem. In the past, a decoder, a search system, was modeled after the left-to-right decoding which produces an output from the beginning of a sentence. This paper presents two alternative directional methods, one is a right-to-left decoding that generates an output from the end of a sentence. The other is a bidirectional decoding that generates an output from both directions. The experimental results of English and Japanese translation indicated that the left-to-right decoding algorithm is good for Japanese-to-English translation, while the right-to-left decoding algorithm was suitable for English-to-Japanese translation. Furthermore, it has been confirmed that the bidirectional decoding algorithm resulted in further reduced search errors.

### 1. はじめに

近年、コーパスを用いた手法の 1 つである統計的機械翻訳システムの研究がさかんに行われるようになってきた。この統計的機械翻訳では、モデルに基づいて翻訳を探索するデコーディングが NP-Complete であることが知られている<sup>1)</sup>。現在まで様々なアルゴリズム、たとえばスタックデコーディングアルゴリズム<sup>2)</sup>、A\*探索アルゴリズム<sup>3),4)</sup>、DP アルゴリズム<sup>5)</sup>、が提案されてきた。これらは、入力文に対して出力文を文頭から生成し、原言語文と目的言語文とがほぼ線形に対応づけられることを仮定したブルーニング戦略を取

り入れたものであった。このようなアルゴリズムにおいては、たとえば日本語と英語のように、語順の異なる言語の翻訳には対応できなかった。また、Germann<sup>6)</sup> は山登り法および線形計画法による解を示している。前者の手法では、満足な解が得られないことが分かっており、また、後者の手法では処理速度の問題で、実用上解を得ることができない。

本稿では、3 つのデコーディングアルゴリズムを取り上げる。まず最初は、幅優先で探索を行い、翻訳を文頭から生成する left-to-right デコーディングを解説する<sup>3)</sup>。次に、翻訳を文頭から生成するのではなく、文末から生成する right-to-left デコーディングを説明し、最後に、文頭、文末両方から翻訳を生成し、双方向から生成された部分文を結合する双方向デコーディングについて説明する。

今回行った実験では、探索エラーの点で日英翻訳では left-to-right デコーディングが良く、英日翻訳では、right-to-left デコーディングの方が良いという結果が

<sup>†</sup> ATR 音声言語コミュニケーション研究所  
ATR Spoken Language Translation Research Laboratories

<sup>††</sup> 京都大学大学院情報学研究科知能情報学専攻  
Department of Intelligence Science and Technology,  
Graduate School of Informatics, Kyoto University

得られた。この結果は、英語は SVO という構造を持ち、文頭から文末へと語順が制約されるのに対し、日本語は SOV であり、文末に語順の制約が強いことから得られたと考えられる。また、双方向デコーディングは、上記の 2 つのアルゴリズムと比べ、探索エラーがさらに減少することが確認された。ただし、訳質の点ではほぼ変わらず、今後の課題として、統計的翻訳モデルの改善が求められる。

次章において、統計的機械翻訳の概要について説明し、特に探索における問題点を取り上げる。3 章では本稿で提唱する right-to-left デコーディングと双方向デコーディングについて概説し、4 章で旅行会話コーパスを用いた実験結果を示す。5 章で実験結果の考察する。

## 2. 統計的機械翻訳

統計的機械翻訳においては、ある原言語のテキスト（たとえば日本語、 $J$ ）を目的言語のテキスト（たとえば英語、 $E$ ）に翻訳する問題を最大尤度の解を発見する問題としてとらえる<sup>7)</sup>。

$$\hat{E} = \operatorname{argmax}_E P(E|J)$$

ベイズルールを適用することにより、上記の問題は通信路モデルにおける最大事後確率の解を求める問題に置き換えられる。

$$\hat{E} = \operatorname{argmax}_E P(E)P(J|E)$$

このため、翻訳とは「ある情報源モデル  $P(E)$  で起こった  $E$  が、雑音のある通進路モデル  $P(J|E)$  により、 $J$  として出力された」と仮定して、 $P(E)$  と  $P(J|E)$  を用いて  $J$  をデコードする、として表現された（図 1 参照<sup>7)</sup>）。

$P(E)$  は言語モデルであり、 $E$  の尤度を示す。 $P(J|E)$  は翻訳モデルであり、ソース文  $E$  がターゲット文  $J$  を生成する確率を表す<sup>1)</sup>。翻訳モデルに単語アライメント  $A$  という概念が導入され、 $E$  と  $J$  との対応関係を  $J$  の各単語がどの  $E$  の単語のインデックスに対応するかで表現された<sup>2,7)</sup>。

$$P(J|E) = \sum_A P(J, A|E)$$

図 2 は、英語のソース文 “Could you recommend another hotel?” とその翻訳、日本語のターゲット文「他のホテルを紹介していただけませんか」のアライメント

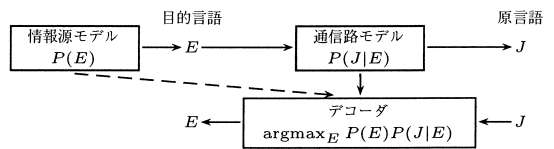


図 1 統計的機械翻訳の枠組み

Fig. 1 Framework for statistical machine translation.

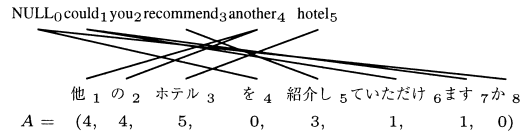


図 2 日本語と英語の文のアライメントの例

Fig. 2 Example of an alignment for Japanese-English sentence.

の例を示している。たとえばターゲットのインデックス 1 の「他」とインデックス 2 の「の」はソースのインデックス 4 の “another” から生成されている。また、ソースのインデックス 0 にある NULL シンボルも単語の 1 つであり、「を」や「か」のように、ターゲット文に対する挿入を表している。

### 2.1 翻訳モデルの構成

翻訳モデル  $P(J, A|E)$  はさらに分解され、以下のように 4 つの過程でモデル化されている<sup>7)</sup>。ここで、 $l, m$  をそれぞれ  $E, J$  の文長とする。

$$P(J, A|E) = \prod_{i=1}^l n(\phi_i|E_i) \times \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \times \prod_{j=1}^m t(J_j|EA_j) \times d(A|J, E)$$

- Fertility モデル:  $n(\phi_i|E_i)$   
まず、ソースのある単語  $E_i$  ( $1 \leq i \leq l$ ,  $E$  の  $i$  番目の要素) が  $\phi_i$  個の単語を生成する<sup>3)</sup>。
- NULL 生成モデル:  $\binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \equiv g(\phi_0|m)$   
ターゲット文の文長  $m$  を満たすため、 $\phi_0$  個の NULL を挿入する必要があり、 $m - \phi_0 (= \sum_{i=1}^l \phi_i)$  個の各単語につき  $p_1 (= 1 - p_0)$  の確率で NULL の挿入を行う<sup>4)</sup>。
- Lexicon モデル:  $t(J_j|EA_j)$   
生成されたソースの各単語  $E_{A_j}$  がターゲットの単語  $J_j$  へと翻訳される。
- Distortion モデル:  $d(A|J, E)$

<sup>1)</sup> ソース/ターゲットは、それぞれ通信路モデルの入力文/出力文であり、翻訳における目的言語文/原言語文、あるいはデコーダの出力文/入力文に相当する。

<sup>2)</sup> ソースの各単語からの複数のターゲット単語への対応が可能。

<sup>3)</sup>  $\phi_i$  を、 $E_i$  の fertility と呼び、0 の場合、消去を表す。

<sup>4)</sup> 二項分布となることに注意。

翻出された単語の並べ替えを行う。Distortion モデルをさらに分解した複雑なモデルが定義されている<sup>7)</sup>。

このように、翻訳は「単語の集合を生成 (Fertility モデル), NULL を挿入 (NULL 生成モデル), 生成された各単語の翻訳 (Lexicon モデル), 翻訳された単語の並べ替え (Distortion モデル)」という過程で表現されている。

2.2 探索問題

統計的機械翻訳における探索は、入力文  $J$  が与えられたときに、翻訳モデル  $P(J|E) = \sum_A P(J, A|E)$  と言語モデル  $P(E)$  から最大尤度の出力文  $\hat{E}$  を発見する。ただし、可能な  $A$  の空間は、 $J$  の単語長を  $m$ ,  $E$  の単語長を  $l$  とすると、 $(l+1)^m$  と非常に大きくなるので、翻訳を決定する場合、その近似値  $P(J|E) \simeq P(J, \hat{A}|E)$  を用いる。つまり、 $J$  が与えられたとき、 $\hat{E} = \operatorname{argmax}_E P(J, \hat{A}|E)P(E)$  となる  $\hat{E}$  を求める問題とする。

この問題は、単語単位の翻訳の選択だけでなく、並べ替えの問題も含んでいることから NP-Complete であることが知られている<sup>1)</sup>。過去の研究においては、出力を文頭から生成する制約を加え、ブルーニングを行うことによりデコーディングアルゴリズムを実現している<sup>2)~6)</sup>。ただし、このブルーニングでは原言語と目的言語とがほぼ線形に対応づけられていることを仮定しており、日本語と英語間のように語順の相違が大きい言語間の翻訳を想定していない。

次章においては、本稿で提唱する出力の方向を考慮したデコーディングアルゴリズムを概説する。

3. デコーディングアルゴリズム

本稿で提唱するデコーディングアルゴリズムは、Tillman ら<sup>5)</sup> や Och ら<sup>3)</sup> によるビームサーチアルゴリズムに基づいており、また、仮説に適用される操作は Berger ら<sup>2)</sup> や Och ら<sup>3)</sup>, Germann ら<sup>6)</sup> の研究に基づいている。

ビームサーチアルゴリズムでは、入力文  $J$  (長さ  $m$ ) に対し、ある翻訳仮説  $(E_1^l, A)$  を仮定する。ここで、 $E_1^l$  を、 $J$  のある単語の部分集合に対応した、長さ  $l$  の翻訳された単語列であり、 $A$  を対応する単語アライメントとする。このような仮説に対して以下のような操作および各仮説に対するスコア  $Q(E_1^l, A)$  を定義することができる (図 3 参照)。

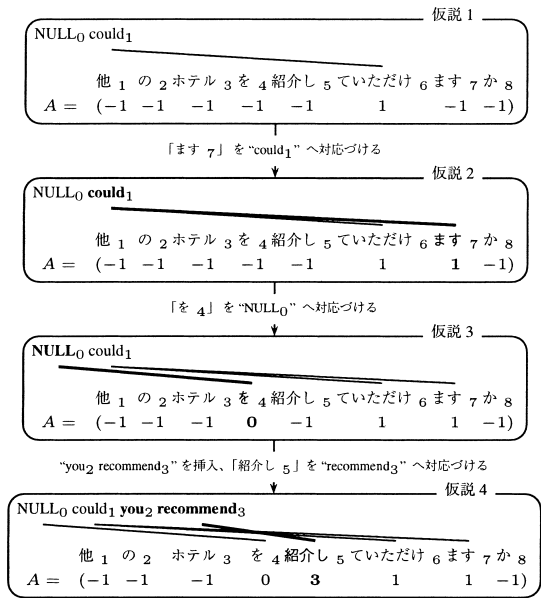


図 3 デコーディング時における操作の例  
Fig. 3 Example of operators during decoding process.

- 単語対応：入力文のうち、アライメントが決定していない単語  $J_j$  を出力文のある単語  $E_i$  に対応づける。たとえば、図 3 の仮説 1 に対し「ます 7」を “could<sub>1</sub>” へ対応づけることにより、仮説 2 を生成可能である。ここで、まだ単語アライメントが決定されていない  $A$  の要素には -1 が割り当てられていると仮定する。この操作では、“could<sub>1</sub>” の fertility が 1 から 2 へと増大し、 $E$  に対しては修正を加えていない。スコア  $Q(E_1^l, A)$  は以下のようなになる。

$$Q(E_1^l, A : A_j = i) = Q(E_1^l, A' : A'_j = -1) \times t(J_j|E_i) \times \frac{n(\phi_i+1|E_i)}{n(\phi_i|E_i)} \times \frac{d(A:A_j=i|J,E_1^l)}{d(A':A'_j=-1|J,E_1^l)}$$

ここで、 $A : A_j = i$  を、 $A_j$  が  $i$  である単語アライメント  $A$  とする。

- NULL 対応：「単語対応」の操作と同様、入力文のある単語  $J_j$  を NULL に対応づける。図 3 では、仮説 2 に対し「を 4」を “NULL<sub>0</sub>” へ対応づけることにより、仮説 3 を生成している。上記の操作と同様、“NULL<sub>0</sub>” の fertility が 1 となっている。スコア  $Q(E_1^l, A)$  は以下のようなになる。

$$Q(E_1^l, A : A_j = 0) = Q(E_1^l, A' : A'_j = -1) \times t(J_j|NULL) \times \frac{g(\phi_0+1|m)}{g(\phi_0|m)} \times \frac{d(A:A_j=0|J,E_1^l)}{d(A':A'_j=-1|J,E_1^l)}$$

- 挿入および対応づけ：出力文のインデックス  $k$  に

+1 は NULL シンボル。  
このときのアライメント  $\hat{A}$  をビターピアアライメントと呼ぶ。

て、ある単語列  $E_1^{l'}$  が挿入され ( $l' \geq 1$ )、出力文が  $E_1^k + E_1^{l'} + E_{k+1}^l \equiv E_1^{k+l'+l}$  となり、入力文の単語  $J_j$  を  $E''_i$  に対応づける ( $k \leq i \leq k+l'$ )。たとえば、仮説 3 に対して “you<sub>2</sub> recommend<sub>3</sub>” を文末に挿入、「紹介<sub>5</sub>」を “recommend<sub>3</sub>” へ対応づけられ、仮説 4 が生成されている。ここで、“you<sub>2</sub>” のように、 $J$  からの対応づけがなされていない単語を zero fertility 単語と呼ぶ。以下のようスコアの計算が行われる。

$$Q(E_1^{k+l'+l}, A: A_j = i) = Q(E_1^l, A': A'_j = -1) \times t(J_j | E''_i) \times \prod_{i'=k+1}^{k+l'} n(\phi_{i'} | E_{i'}) \times \frac{d(A: A_j = i | J, E_1^{k+l'+l})}{d(A': A'_j = -1 | J, E_1^l)} \times \frac{P_{lm}(E_1^{k+l'+l})}{P_{lm}(E_1^l)}$$

このように、デコーダは上記の操作を行い、スコア  $Q$  が最大となる仮説 ( $E_1^l, A$ ) を得る問題となる。実装上では、幅優先で探索を行い、入力文  $J$  のうち処理された単語数に同期して処理が進む (図 4 参照)。 $c$  を処理された入力単語長とし、各  $c$  に対応した、仮説のスコアによる優先度つきキュー  $B_c$  が存在すると仮定する ( $0 \leq c \leq m$ )。まず、 $B_c$  に翻訳を行っていない初期状態 ( $E_0^0, A = (-1, -1, \dots, -1)$ ) を入れる (図 4, 行番号 3)。その後、各  $B_c$  ( $0 \leq c \leq m-1$ ) の各仮説に対し、上記のような操作を加え、新たな仮説を生成し、 $B_{c+1}$  へと挿入を行う (図 4, 行番号 6 から 12)。そして、 $B_m$  から最もスコアの高い仮説を取り出すことにより、デコードが終了する。ここで、ある仮説 ( $E_1^l, A$ ) において、すでに処理された入力文の単語の位置の集合を  $C$  とすると、図 4 で示されたアルゴリズムの探索空間は、図 5 のようにラティスで表すことができる。また、各  $C$  の大きさに対応した  $B_{|C|}$  が存在することが分かる。

図 4, 行番号 9 では、ある単語列  $E_1^{l'}$  の生成が行われている。本稿では、すべての可能な  $E_1^{l'}$  を列挙するのではなく、逆 Lexicon モデル<sup>2)</sup>  $P(E_i | J_j) = P(J_j | E_i)P(E_i) / \sum_{E_i} P(J_j | E_i)P(E_i)$  とトレーニングコーパスのビターアライメントから得られる zero fertility 単語列との組合せで候補を求めている<sup>8)</sup>。

デコーディング中に仮説の爆発を避けるため、 $B_c$  の大きさを制限する、あるいは、キューの中で最大のスコアよりも一定の割合スコアが低い仮説を取り除くといった、ビームプルーニングを行う。また、「単語対応」の操作により出力単語の fertility が増大するが、

```

1: 入力単語列:  $J_1, J_2, \dots, J_m$ 
2: 出力 翻訳仮説の集合:  $B_m$ 
3: 初期化: ( $E_0^0, A = (-1, -1, \dots, -1)$ ) を  $B_0$  に入れる
4: for all  $c = 0, 1, \dots, m-1$  do
5:   for all ( $E_1^l, A$ ) in  $B_c$  do
6:     for all  $j = 1, \dots, m$  and  $A_j = -1$  do
7:       任意の  $E$  のインデックス  $i$  に対し、 $J_j$  を  $E_i$  に対応づけ、
       ( $E_1^l, A'$ ) を生成、 $Q(E_1^l, A')$  を計算し、 $B_{c+1}$  へ挿入
8:        $J_j$  を NULL に対応づけ、( $E_1^l, A'$ ) を生成、 $Q(E_1^l, A')$  を
       計算し、 $B_{c+1}$  へ挿入
9:     for all 単語列:  $E_1^{l'}$  do
10:      任意の  $E$  のインデックス  $i$  に対し、単語列  $E_1^{l'}$  を挿入、 $J_j$ 
       を  $E_i$  に対応づけ、( $E_1^{l'+l}, A'$ ) を生成、 $Q(E_1^{l'+l}, A')$ 
       を計算し、 $B_{c+1}$  へ挿入
11:     end for
12:   end for
13: end for
14: end for
    
```

図 4 ビームサーチアルゴリズム

Fig. 4 Beam search algorithm.

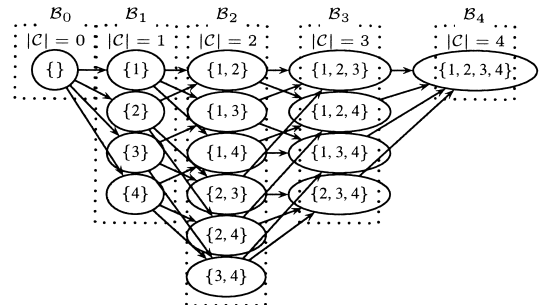


図 5 入力単語長が 4 の場合の探索空間

Fig. 5 Search space for the input length of 4.

fertility の大きさを制約する fertility プルーニングを取り入れた<sup>8)</sup>。Och ら<sup>3)</sup> によるアライメントを制約する手法、skip ベースの制約が存在するが、日本語と英語の翻訳においては線形的なアライメントを仮定できないため、本稿では用いていない。

上記のアルゴリズムでは、出力方向についての制約を規定しておらず、各操作において「任意の  $i$ 」に対して出力単語への対応が行われている。次節以降において、出力方法に制約を加えた、2 種類のアルゴリズム、left-to-right と right-to-left のアルゴリズムを示す。また、両方向にデコーディングを行う、双方向デコーディングも説明する。

### 3.1 Left-to-Right デコーディング

left-to-right デコーディングにおいて、「単語対応」の操作は、単語  $J_j$  は  $E_1^l$  の文末の単語  $E_l$  への対応づけが行われる ( $i = l$ , 図 6 (a) 参照)。また「挿入および対応づけ」の操作では、単語列  $E_1^{l'}$  の挿入が  $E_1^l$  の最後に行われ、入力文の  $J_j$  からの単語アライメントは  $E_1^{l'+l}$  へ対応づけられる (図 6 (b) 参照)。

### 3.2 Right-to-Left デコーディング

left-to-right が文頭から部分文を生成するのに対し

A に -1 以外の値が埋められた位置の集合。

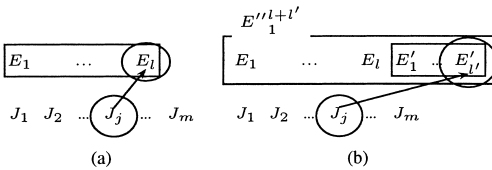


図 6 left-to-right デコーディングにおける単語列の挿入, および入力語の対応づけ  
Fig. 6 Insertion and alignment for left-to-right decoding.

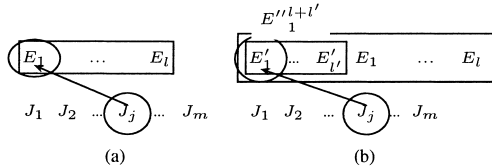


図 7 right-to-left デコーディングにおける単語列の挿入, および入力語の対応づけ  
Fig. 7 Insertion and alignment for right-to-left decoding.

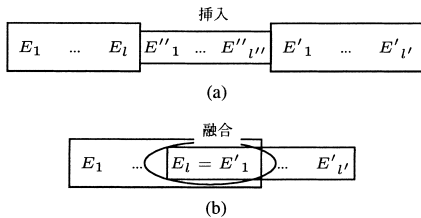


図 8 left-to-right, right-to-left の仮説の結合  
Fig. 8 Merge of left-to-right and right-to-left hypothesis.

て, right-to-left デコーディングでは文末から翻訳を生成する「単語対応」の操作は, 単語  $J_j$  は  $E_1^l$  の文頭の単語  $E_1$  への対応づけが行われる ( $i = 1$ , 図 7(a) 参照). また「挿入および対応づけ」の操作では, 単語列  $E_1^{l'}$  の挿入が  $E_1^l$  の先頭に行われ, 入力文の  $J_j$  からの単語アライメントは  $E_1^{l'}$  へ対応づけられる (図 7(b) 参照). ただし,  $A$  のインデックスの割当てを再度行う.

3.3 双方向デコーディング

双方向デコーディングでは, 入力文に対して left-to-right, right-to-left 同時にデコーディングを行い, 両方向からの仮説を結合することにより出力を生成する. 本稿では,  $m$  を入力長とすると, left-to-right 方向に  $\lceil m/2 \rceil$  単語, および right-to-left 方向に  $\lfloor m/2 \rfloor$  単語それぞれ処理を行う. left-to-right, right-to-left の処理による仮説をそれぞれ  $(E_1^l, A)$ ,  $(E_1^{l'}, A')$  とし, 各仮説に対応する, すでに処理された入力文の単語の位置の集合を  $C, C'$  とすると,  $C \cap C'$  が空集合の場合に結合可能である. また, 結合する場合,  $E_1^l$  と  $E_1^{l'}$  の間に単語列  $E_1^{l''}$  の挿入を行う (図 8(a) 参照). あるいは,  $E_l = E'_1$  の場合,  $E_l$  と  $E'_1$  とを融合し, 新しい

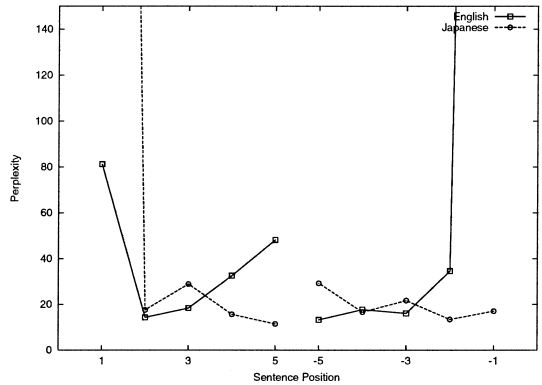


図 9 日英言語モデルのパープレキシティー  
Fig. 9 Language model perplexity for Japanese and English.

出力  $E_1^{l-1} + E_1^{l'}$  を形成可能である (図 8(b) 参照).

3.4 計算量

left-to-right と right-to-left のデコーディングアルゴリズムの計算量は等しく, Tillmannら<sup>5)</sup>が指摘したように,  $|E|$  を出力言語の語彙の数,  $m$  を入力長とすると,  $O(|E|^3 m^2 2^m)$  である. 双方向デコーディングの場合, 双方向からの仮説を結合する必要があることから,  $O(\binom{m}{m/2})$  がさらに必要となる.

このように, 非常に大きい探索空間の中から最適な翻訳を発見する必要があり, プルーニングが必要となるが, 探索エラーが起きる可能性が大きくなる.

3.5 デコーディング方向と目的言語の特性

従来の left-to-right デコーディングにおいては, 入力単語を処理し, 対応する翻訳を単語単位に決定して出力を文頭から生成する. すなわち, 言語モデルや翻訳モデルは文頭からスコアリングを行うことにより, 「良い仮説」と「悪い仮説」とを区別している. たとえば, 英語やドイツ語, フランス語のように文頭からの語順の制約が強い言語を目的語として翻訳を行った場合, 文頭からのスコアリングによる制約が効いて, ビームサーチのプルーニングを行っても出力の質が落ちることがない. 逆に, このような言語に対して right-to-left にデコーディングを行うと, 仮説間のスコアの差が小さく, 仮説数が増大し, プルーニングによる正解のれれが大きくなる.

同様に, 日本語のような, 文末に語順の制約がある場合, そのような言語を目的語として right-to-left デコーディングを行うと, スコアの差が大きくなるが, left-to-right に行くとスコアの差が小さくなると考えられ, right-to-left にデコーディングを行うことが優

$|E|^3$  は 3-gram 言語モデルを使用した場合.

位になる．

図 9 に，left-to-right，right-to-left それぞれのデコーディングで用いる 3-gram 言語モデルの文の位置による英語，日本語の平均パープレキシティを示す（コーパス等実験条件は 4 章参照）．ある文を  $W_1^n$  とすると，left-to-right の言語モデルは  $P(W_1^n) = \prod_{k=1}^n P(W_k|W_{k-2}, S_{k-1})$  で表され，right-to-left の言語モデルは  $P(W_1^n) = \prod_{k=1}^n P(W_k|W_{k+1}, S_{k+2})$  を用いる．正の位置（1 から 5）は文頭からの位置，負の位置（-1 から -5）は文末からの位置であり，それぞれ，left-to-right，right-to-left の言語モデルによるパープレキシティを示す．図 9 から，日本語は英語と比べて文頭のパープレキシティが非常に高く，制約がゆるいと考えられるが，文末ではほぼ同じようなパープレキシティであった．また，英語は文末でのパープレキシティが非常に高く，逆に制約が非常に緩いことが分かる．

双方向デコーディングの場合，両者の利点を取り入れていることから，どのような言語に対しても両者の利点を取り入れた翻訳結果になることが期待される．図 5 に示された探索空間では，入力単語を処理した数  $c$  が入力文の半分の大きさになったとき，仮説の組合せが最も多く， $\binom{m}{m/2}$  となり，大幅に仮説数が増大する．単方向で処理を行った場合，仮説数が多くなり，これ以降の探索でエラーが起きる可能性が大きくなる．双方向デコーディングでは，この状態のときに仮説の結合を行うことから，プルーニングによる影響が少なくなる．

次章では，日本語と英語を対象として翻訳実験を行った結果を示す．

#### 4. 実 験

今回行った実験のコーパスは，表 1 に示された，旅行会話基本表現集（BTEC<sup>9)</sup> から抽出した 172,481 の日英の対訳文である．このコーパスから 152,183 文からなるトレーニングセット，検証セット 10,148 文，さらにテストセット 10,150 文を取り出した．

翻訳モデルは日英，英日方向，どちらも IBM Model 1 から IBM Model 4 までトレーニングを行い，途中で HMM Model によるトレーニングを行っている<sup>10)</sup>．IBM Model 4 においては，品詞を単語クラスとして用いた．

トレーニングコーパスのビターピアライメントから，頻度が 10 以上の zero fertility 単語列を取り出した．

表 1 旅行会話基本表現集（BTEC）  
Table 1 Basic travel expression corpus.

	日本語	英語
文数	172,481	
単語数	1,186,620	1,005,080
語彙数	22,801	15,768
平均文長	6.88	5.83
3-gram パープレキシティ	26.16	36.92

また，テストセットから文長が 6，8，10 である，それぞれ 50 文，合計 150 の日本語，英語の文を取り出し，上記の 3 つのアルゴリズムによるデコーディングの実験を行った．

翻訳結果を評価するにあたって，以下の基準を用いた．

- 単語誤り率（WER）：挿入，削除，置換に対してペナルティを 1 として計算．
- 位置独立単語誤り率（PER）：WER では，実際の翻訳に起こりうる並べ替えに対してペナルティが大きくなる．したがって，PER では位置の情報を無視し，出力単語の集合の正しさの評価を行う<sup>3)</sup>．
- BLEU スコア：正解集合のすべてに対して N-gram の正解率を計算するものであり，短い出力に対してペナルティを与えている<sup>11)</sup>．
- NIST スコア：BLUE と同様なスコアであるが，正解率を正解集合の N-gram に対する平均出力単語情報量として表現している<sup>12)</sup>．
- 主観評価（SE）：翻訳結果を人手で A，B，C，D と 4 段階にランク付けをする．
- 探索誤り率（SER）：翻訳の正しさあるいは誤り以外に，各デコーディングアルゴリズムが翻訳モデルと言語モデルから最適解を得られたかどうかを評価した．探索空間が非常に大きいことから，本稿では，各アルゴリズムの出力結果のうち，スコアが最も高い出力の集合を正解集合として，誤り率を求めた．

表 2 に，left-to-right (LtoR) と right-to-left (RtoL)，双方向 (Bi) のデコーディングの翻訳結果を示す．日英，英日の翻訳例をそれぞれ図 10，図 11 に示す．

ここで用いられた評価基準は，A — 完全訳：含まれる情報，文法的に問題なし；B — 部分訳：一部に重要でない情報の欠落や，文法的欠陥があるがすぐに理解できる；C — 可能訳：かなりの情報の欠落があるが，努力すれば理解できる；D — 不可能訳：重要な情報が誤訳されている；である．

句点，読点等の記号はあらかじめ取り除いた．

表 2 日英翻訳, 英日翻訳結果

Table 2 Translation results for Japanese-English/English-Japanese.

翻訳	アルゴリズム	SER [%]	WER [%]	PER [%]	BLEU [%]	NIST	SE [%]			
							A	A+B	A+B+C	D
日英	LtoR	14.0	69.7	65.3	24.0	4.57	46.0	68.0	82.0	18.0
	RtoL	24.0	70.3	65.0	23.2	4.54	44.0	66.7	82.7	17.3
	Bi	11.3	70.1	65.4	23.3	4.54	46.3	68.3	80.7	19.3
英日	LtoR	13.3	66.3	58.3	25.4	4.39	44.7	57.4	68.0	32.0
	RtoL	12.0	66.2	57.6	25.5	4.39	44.0	55.3	65.3	34.7
	Bi	9.3	66.6	59.0	25.3	4.34	42.7	54.7	66.0	34.0

LtoR: left-to-right デコーディング  
 RtoL: right-to-left デコーディング  
 Bi: 双方向デコーディング

入力:	スリに財布をすられました
LtoR:	here 's my wallet was stolen
RtoL:	here 's my wallet was stolen
Bi:	i had my wallet stolen
入力:	すみません が テラスの 席が いいの ですが
LtoR:	excuse me i 'd like a seat on the terrace
RtoL:	i 'd prefer excuse me
Bi:	i 'd like a seat on the terrace
入力:	何時に 終る の です
LtoR:	what time should i be at the end
RtoL:	it 's what time will it be over
Bi:	at what time is it end
入力:	荷物を 上 に あげ て もらえ ます か
LtoR:	could you put my baggage here
RtoL:	do you have overhead luggage
Bi:	could you put my baggage
入力:	ええ 兄 と 妹 が 1 人 ず つ います
LtoR:	yes brother and sister there a daughter
RtoL:	you 're yes brother and sister daughter
Bi:	yes my daughter is there a brother and sister

図 10 日英翻訳の例

Fig. 10 Example of Japanese-to-English translation.

## 5. 考 察

表 2 における探索誤り率 (SER) から, 日英翻訳については right-to-left デコーディングより left-to-right デコーディングが良く, また, 英日翻訳では, その逆の結果が得られた. このことから, 言語モデル, 翻訳モデルから翻訳結果を得るにあたって, 出力文の言語の特性に合わせて出力の方向を制約する手法の有効性が確認された. また, 双方向デコーディングではさらにエラー率が低かった. これは, 文頭, 文末の制約を同時に用いることにより, 仮説がビームより落ちることが少なくなったと考えられる.

ただし, 翻訳結果の客観評価値 (WER/PER/BLEU/NIST) あるいは主観評価値 (SE) において, 日英翻訳, 英日翻訳ともにアルゴリズム間に優位な差異があまり見られなかった.

図 10 や図 11 で示された翻訳例のように, 統計的機械翻訳はまだまだ実用化に至る技術ではなく, 課題が多いことが分かる. 英日, 日英翻訳ともに, 出力言語に合わせて生成方向を変化させることにより, 探索

入力:	this is my boss mr. kitano
LtoR:	これは私の上司であります
RtoL:	北野 上司 です
Bi:	これは私の上司であります
入力:	i 'm glad that we have a hobby in common
LtoR:	私の趣味が 痛いと 言って うれしい です
RtoL:	私の趣味熱が あって うれしい です
Bi:	私の趣味が あって うれしい です
入力:	excuse me sir what do you want for drink before dinner
LtoR:	すみません が 食事の 前に 何か 飲み物 を お 願 い し ます か
RtoL:	いくら になり ます 食事の 前に 飲み物 が 欲 しい の です か
Bi:	すみません いくら になり ます 食事の 前に 飲み物 が 欲 しい の です か
入力:	sorry but i don't understand these medical words in english
LtoR:	すみません この 言葉 は 英語 が 分 かり ませ ん
RtoL:	診断書 あいにく この 言葉 は 英語 が よく 分 かり ませ ん
Bi:	すみません この 言葉 は 英語 が 分 かり ませ ん
入力:	hurry up or i 'll be late for the flight
LtoR:	その 便 は どの くらい 遅れ ます か それとも 急い て くだ さい
RtoL:	飛行機 の 急い て います か それとも 遅れ ます
Bi:	その 便 の 急い て います か それとも 遅れ ます

図 11 英日翻訳例

Fig. 11 Example of English-to-Japanese translation.

エラー率が減少する結果が見られたが, 客観評価, 主観評価による訳質評価では, ほぼ変わらない結果が得られている. これは, IBM Model 自体が比較的近い言語を想定しており<sup>7)</sup>, 日英等, 異なる言語対を想定していないことから, 探索エラー率の減少がそのまま翻訳性能に寄与していないことが考えられる.

## 6. おわりに

本稿では, 統計的機械翻訳におけるデコーディングアルゴリズムでは, 従来, 文頭から目的言語文を生成する left-to-right デコーディングが用いられてきたが, 文末から生成する right-to-left デコーディングを示した. さらに, 双方向に生成する双方向デコーディングを提案した. 英語, 日本語の翻訳実験結果から, 探索誤り率の点では日英翻訳では left-to-right デコーディングが良く, 英日翻訳では right-to-left デコーディン

グの手法が良いことが確認された。ただし、訳質の点では変化がみられず、今後の課題としては、日英両言語の特性を考慮したモデル化を行う必要があると考えられる<sup>13)</sup>。また、探索アルゴリズムを改善する手法が考えられ、たとえば、用例を用いることにより、非常に大きい探索空間にバイアスを加えて探索を行う手法が考えられる<sup>14)</sup>。

謝辞 本研究は通信・放送機構の研究委託「大規模コーパスベース音声対話翻訳技術の研究開発」により実施したものである。本稿を完成するにあたり、様々な助言をくださった京都大学河原教授に感謝いたします。また、様々な議論をしてくださった、ATR 音声言語コミュニケーション研究所山本所長、および前白井室長、中岩室長に感謝いたします。

### 参 考 文 献

- 1) Knight, K.: Decoding Complexity in Word-Replacement Translation Models, *Computational Linguistics*, Vol.25, No.4, pp.607-615 (1999).
- 2) Berger, A.L., Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Kehler, A.S. and Mercer, R.L.: Language Translation Apparatus and Method of Using Context-Based Translation Models, Technical report, United States Patent, Patent Number 5510981 (1996).
- 3) Och, F.J., Ueffing, N. and Ney, H.: An Efficient A\* Search Algorithm for Statistical Machine Translation, *Proc. ACL-2001 Workshop on Data-Driven Machine Translation*, Toulouse, France, pp.55-62 (2001).
- 4) Wang, Y.-Y. and Waibel, A.: Decoding Algorithm in Statistical Machine Translation, *Proc. ACL 1997* (1997).
- 5) Tillmann, C. and Ney, H.: Word Re-ordering and DP-based Search in Statistical Machine Translation, *Proc. COLING 2000* (2000).
- 6) Germann, U., Jahr, M., Knight, K., Marcu, D. and Yamada, K.: Fast Decoding and Optimal Decoding for Machine Translation, *Proc. ACL 2001*, Toulouse, France (2001).
- 7) Brown, P.F., Pietra, S.A.D., Pietra, V.J.D. and Mercer, R.L.: The Mathematics of Statistical Machine Translation: Parameter Estimation, *Computational Linguistics*, Vol.19, No.2, pp.263-311 (1993).
- 8) Watanabe, T. and Sumita, E.: Bidirectional Decoding for Statistical Machine Translation, *Proc. COLING 2002*, Taipei, Taiwan, pp.1079-1085 (2002).
- 9) Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H. and Yamamoto, S.: Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World, *Proc. LREC 2002*, Las Palmas, Canary Islands, Spain, pp.147-152 (2002).
- 10) Och, F.J. and Ney, H.: Improved Statistical Alignment Models, *ACL 2000*, Hongkong, China, pp.440-447 (2000).
- 11) Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation, *Proc. ACL 2002*, pp.311-318 (2002).
- 12) Automatic Evaluation of Machine Translation Quality using N-gram Co-Occurrence Statistics, <http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf> (2002).
- 13) Watanabe, T., Sumita, E. and Okuno, H.G.: Chunk-based Statistical Translation, *Proc. ACL 2003*, Sapporo, Japan (2003).
- 14) Marcu, D.: Towards a Unified Approach to Memory and Statistical-Based Machine Translation, *Proc. ACL 2001*, Toulouse, France, pp.378-385 (2001).

(平成 14 年 9 月 13 日受付)

(平成 15 年 10 月 16 日採録)



渡辺 太郎

1994 年京都大学工学部情報工学科卒業。1997 年京都大学大学院工学研究科情報工学専攻修士課程修了。同年、京都大学工学研究科情報工学専攻後期博士課程入学。2000 年 Language and Information Technologies, School of Computer Science, Carnegie Mellon University, Master of Science 取得。現在、ATR 音声言語コミュニケーション研究所研究員。機械学習、言語処理、統計的機械翻訳の研究。言語処理学会会員。



隅田英一郎 (正会員)

1982 年電気通信大学大学院計算機科学専攻修士課程修了。1999 年京都大学工学博士。ATR 音声言語コミュニケーション研究所主任研究員。自然言語処理、機械翻訳、情報検索、並列処理の研究に従事。電子情報通信学会、ACL 各会員。

**奥乃 博**

1972年東京大学教養学部基礎科学科卒業。日本電信電話公社，NTT，科学技術振興事業団北野共生システムプロジェクト，東京理科大学理工学部情報科学科を経て，2001年4

月より京都大学大学院情報学研究科知能情報学専攻教授。博士（工学）。この間，スタンフォード大学客員研究員，東京大学工学部客員助教授。人工知能，音環境理解，ロボット聴覚，音楽情報処理の研究に従事。IEA/AIE-2001 最優秀論文賞，IROS-2001 Best Paper Nomination Finalist，第2回船井情報科学振興賞等受賞。本学会英文図書委員。人工知能学会，日本認知科学会，日本ソフトウェア科学会，日本ロボット学会，ACM，AAAI，ASA 各会員。著編書：『インターネット活用術』（岩波書店，1996），“Computational Auditory Scene Analysis”（共編，LEA，1998），“Advanced Lisp Technology”（共編，Francis & Taylor，2002）等。

---