

## 概要

機械翻訳の問題点の1つとして、入力文中の単語が未知語として出現する問題点がある。この対策として川原ら [1] は、IBM Model1 を用いて未知語を翻訳する手法（以下、未知語処理）を提案した。この手法は、対訳学習文と IBM Model1 をもとに作成した対訳単語辞書を用いて、未知語処理を翻訳した。しかし、この手法にはまだ課題がある。未知語処理の課題の一つとして対訳単語辞書が挙げられる。使用した対訳単語辞書の精度が低いことが、翻訳精度の低下を招いていると考える。

未知語処理に使用している、対訳単語辞書を調査する研究がある。中村ら [2] は、対訳単語辞書の日本語単語と英語単語の適切な対応の数を調査した。しかし、この研究では GIZA++[6] の対訳単語確率の計算において IBM 翻訳モデル [3] の Model1 のみの調査で、他のモデルでの調査が行われていない。GIZA++の対訳単語確率を計算するモデルとしては、IBM Model1 ~ Model5 や HMM[4] などがある。また未知語の多くは、対訳学習文中に1回のみ出現する単語（以下、頻度1単語）である。

そこで本研究では、IBM Model1 ~ Model5 と HMM において頻度1単語の精度を調査する。具体的には、対訳学習文と GIZA++の対訳単語確率（IBM Model1 ~ Model5 と HMM）を用いて対訳単語を作成する。作成した対訳単語から枝刈りを行い、対訳単語辞書を作成する。そして対訳単語辞書に含まれる頻度1単語の精度を調査する。結果として、頻度1単語の精度はどのモデルも大きな差はなかった。

# 目次

第1章	はじめに	1
第2章	従来手法	2
2.1	未知語処理	2
2.2	対訳単語辞書	4
2.3	IBM 翻訳モデル	6
2.3.1	Model1	7
2.3.2	Model2	8
2.3.3	Model3	9
2.3.4	Model4	10
2.3.5	Model5	10
2.4	HMM	11
2.5	GIZA++	15
2.6	中村らの研究 [2]	16
2.6.1	概要	16
2.6.2	問題点	16
第3章	調査手法	17
3.1	調査手法の概要	17
3.2	研究の目的	17
第4章	実験	18
4.1	実験データ	18
4.2	調査条件	18
4.3	枝刈り条件	19
4.4	評価方法	19

<b>第 5 章</b>	<b>実験結果</b>	<b>20</b>
5.1	評価結果	20
5.2	頻度 1 単語の種類別の評価	21
5.3	作成した対訳単語辞書に含まれる例	22
<b>第 6 章</b>	<b>考察</b>	<b>30</b>
6.1	調査対象の変更	30
6.1.1	作成した対訳単語辞書の全単語	30
6.1.1.1	評価結果	30
6.1.1.2	作成した対訳単語辞書に含まれる例	31
6.1.2	頻度 2 以上の単語	34
6.1.2.1	評価結果	34
6.1.2.2	作成した対訳単語辞書に含まれる例	35
6.2	学習回数の変更	38
6.2.1	評価結果	38
6.2.2	頻度 1 単語の種類別の評価	39
6.2.3	作成した対訳単語辞書に含まれる例	40
6.3	順位の枝刈り条件の変更	43
6.3.1	日本語順位 1 位・英語順位 8 位以内	43
6.3.1.1	評価結果	43
6.3.1.2	作成した対訳単語辞書に含まれる例	44
6.3.2	日本語・英語ともに順位 1 位	46
6.3.2.1	評価結果	46
6.3.2.2	作成した対訳単語辞書に含まれる例	47
6.4	未知語の翻訳	50
6.4.1	実験設定	50
6.4.2	使用データ	50
6.4.3	自動評価	51
6.4.4	人手評価	52
6.4.4.1	人手評価方法	52
6.4.4.2	人手評価結果	52
6.4.4.3	人手評価例：未知語処理 の例	53

6.4.4.4	人手評価例：未知語処理 × の例 . . . . .	54
6.4.4.5	人手評価例：差なしの例 . . . . .	55

第7章	おわりに	56
-----	------	----

# 図目次

2.1 未知語処理の流れ図 . . . . .	3
2.2 対訳単語辞書作成の流れ図 . . . . .	5

# 表 目 次

2.1.1 未知語を含む出力文 . . . . .	2
2.1.2 対訳単語辞書 . . . . .	2
2.1.3 未知語処理後の出力文 . . . . .	3
2.2.1 単語対応の取得 . . . . .	4
4.1.1 単文コーパスの例 . . . . .	18
4.4.1 評価基準 . . . . .	19
5.1.1 評価結果 . . . . .	20
5.2.1 頻度 1 単語の種類別の評価結果 . . . . .	21
5.3.1 作成した対訳単語辞書に含まれる日本語単語「連合」の全ての例 . . . . .	22
5.3.2 作成した対訳単語辞書に含まれる日本語単語「備品」の全ての例 . . . . .	23
5.3.3 作成した対訳単語辞書に含まれる日本語単語「長寿」の全ての例 . . . . .	24
5.3.4 作成した対訳単語辞書に含まれる日本語単語「ふた」の全ての例 . . . . .	25
5.3.5 作成した対訳単語辞書に含まれる日本語単語「アグネス」の全ての例 . . . . .	26
5.3.6 作成した対訳単語辞書に含まれる日本語単語「1932」の全ての例 . . . . .	27
5.3.7 作成した対訳単語辞書に含まれる日本語単語「ISO」の全ての例 . . . . .	28
5.3.8 作成した対訳単語辞書に含まれる日本語単語「=」の全ての例 . . . . .	29
6.1.1 全単語の評価結果 . . . . .	30
6.1.2 作成した対訳単語辞書に含まれる日本語単語「尽力」の例の一部 . . . . .	31
6.1.3 作成した対訳単語辞書に含まれる日本語単語「本土」の例の一部 . . . . .	32
6.1.4 作成した対訳単語辞書に含まれる日本語単語「いか」の例の一部 . . . . .	33
6.1.5 頻度 2 以上の単語の評価結果 . . . . .	34
6.1.6 作成した対訳単語辞書に含まれる日本語単語「いくぶん」の例の一部 . . . . .	35
6.1.7 作成した対訳単語辞書に含まれる日本語単語「イナゴ」の例の一部 . . . . .	36
6.1.8 作成した対訳単語辞書に含まれる日本語単語「流入」の例の一部 . . . . .	37

6.2.1 学習回数が2,4,8,16回の評価結果 . . . . .	38
6.2.2 頻度1単語の種類別の評価結果 . . . . .	39
6.2.3 作成した対訳単語辞書に含まれる日本語単語「かまぼこ」の全ての例 . . . . .	40
6.2.4 作成した対訳単語辞書に含まれる日本語単語「スロベニア」の全ての例 . . . . .	41
6.2.5 作成した対訳単語辞書に含まれる日本語単語「贈呈」の全ての例 . . . . .	42
6.3.1 日本語順位1位のみ・英語順位8位以内の評価結果 . . . . .	43
6.3.2 作成した対訳単語辞書に含まれる日本語単語「両生類」の全ての例 . . . . .	44
6.3.3 作成した対訳単語辞書に含まれる日本語単語「脛骨」の全ての例 . . . . .	45
6.3.4 作成した対訳単語辞書に含まれる日本語単語「異名」の全ての例 . . . . .	45
6.3.5 日本語・英語ともに順位が1位のみでの評価結果 . . . . .	46
6.3.6 作成した対訳単語辞書に含まれる日本語単語「叱責」の全ての例 . . . . .	47
6.3.7 作成した対訳単語辞書に含まれる日本語単語「珍重」の全ての例 . . . . .	48
6.3.8 作成した対訳単語辞書に含まれる日本語単語「帝劇」の全ての例 . . . . .	49
6.4.1 未知語処理 の例 . . . . .	50
6.4.2 自動評価結果 . . . . .	51
6.4.3 翻訳実験の人手評価結果 . . . . .	52
6.4.4 未知語処理 の例1 . . . . .	53
6.4.5 未知語処理 の例2 . . . . .	53
6.4.6 未知語処理 の例3 . . . . .	53
6.4.7 未知語処理 × の例1 . . . . .	54
6.4.8 未知語処理 × の例2 . . . . .	54
6.4.9 未知語処理 × の例3 . . . . .	54
6.4.10 差なしの例1 . . . . .	55
6.4.11 差なしの例2 . . . . .	55
6.4.12 差なしの例3 . . . . .	55

# 第1章 はじめに

機械翻訳において，入力文中の単語が未知語として出現する．この未知語を翻訳する手法（以下，未知語処理）が，数多く提案されている [1]．しかし，未知語処理にはまだ課題が多い．

未知語処理の1つとして，対訳学習文と対訳単語確率をもとに作成した対訳単語辞書を用いる手法がある [1]．そして，対訳単語辞書の精度を調査する研究がある [2]．中村らの研究 [2] では，対訳単語辞書の日本語単語と英語単語の適切な対応の数を調査した．この研究の問題点は，主に2点ある．1点目は，GIZA++ [6] の対訳単語確率の計算において IBM 翻訳モデル [3] の Model1 のみの調査で，他のモデルでの調査が行われていない．2点目は，対訳単語辞書に含まれる全ての対訳単語を調査対象としており，未知語処理を対象としていない．対訳単語確率を計算するモデルとしては，IBM Model1 ~ Model5 や HMM [4] などがある．また未知語の多くは，対訳学習文中に1回のみ出現する単語（以下，頻度1単語）である．

そこで本研究では，IBM Model1 ~ Model5 と HMM において頻度1単語の精度を調査する．具体的には，対訳学習文と GIZA++ の対訳単語確率（IBM Model1 ~ Model5 と HMM）を用いて対訳単語を作成する．作成した対訳単語から枝刈りを行い，対訳単語辞書を作成している．そして対訳単語辞書に含まれる頻度1単語の精度を調査する．

本論文の構成を以下に示す．第2章で従来手法について説明し，第3章で調査手法について説明する．そして第4章で実験について，第5章で実験結果を示す．そして，第6章では本研究の考察を述べる．



## 第2章 従来手法

### 2.1 未知語処理

機械翻訳には，未知語処理の手法が数多く提案されている．本節では，本論文で扱う未知語処理の流れ [1] を図 2.1 に示す．また，具体的な手順を以下に示す．

#### 手順 1 未知語を含む文の生成

任意の機械翻訳機より，未知語を含む文を出力する．未知語を含む出力文の例を表 2.1.1 に示す．

表 2.1.1: 未知語を含む出力文

入力文	彼は誤植を見つけた．
出力文	He found a <u>誤植</u> .

#### 手順 2 未知語を翻訳

手順 1 で出力された未知語を対訳単語辞書を用いて翻訳する．訳語が複数存在する場合は，対訳単語確率が最大のものを選択する．なお，表 2 の  $\log_2(P(E|J))$  は，日本語単語が英語単語に訳される GIZA++ [6] の対訳単語確率である．対訳単語辞書の例を表 2.1.2 に示す．

表 2.1.2: 対訳単語辞書

日本語	英語	$\log_2(P(E J))$
本	book	-0.297
誤植	misprint	-3.816

### 手順3 未知語処理後の出力文の生成

未知語処理後の出力文を生成する．未知語処理後の出力文の例を表 2.1.3 に示す．

表 2.1.3: 未知語処理後の出力文

入力文	彼は誤植を見つけた．
出力文	He found a <u>misprint</u> .

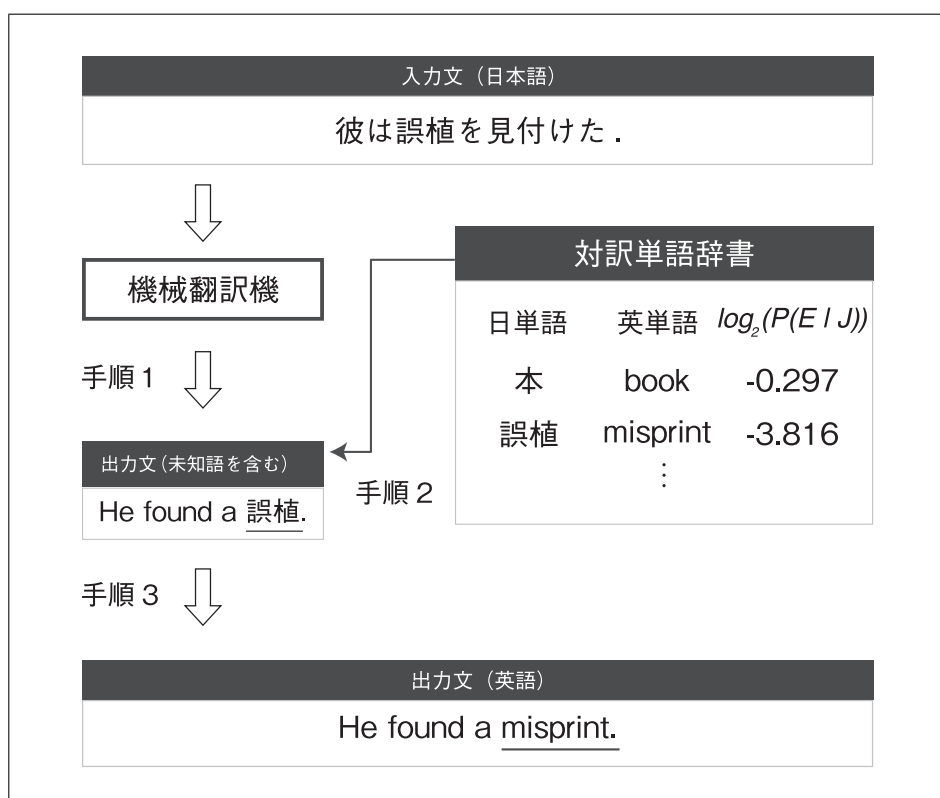


図 2.1: 未知語処理の流れ図

## 2.2 対訳単語辞書

2.1 節の手順 2 で使用している対訳単語辞書は，対訳学習文と GIZA++ を用いて，対訳単語に対訳単語確率を付与して作成する [2]．対訳単語辞書の作成の流れを図 2.2 に示す．また，具体的な手順を以下に示す．

### 手順 1 単語対応の取得

対訳学習文と GIZA++ から，日英方向の単語対応と英日方向の単語対応を取得する．なお，表 4 の  $\log_2(P(J|E))$  は，英語単語が日本語単語に訳される GIZA++ の対訳単語確率である．単語対応の取得の例を表 2.2.1 に示す．

表 2.2.1: 単語対応の取得

日本語単語	英語単語	$\log_2(P(E J))$	$\log_2(P(J E))$
本	book	-0.297	-0.144
誤植	misprint	-3.816	-1.824

### 手順 2 枝刈り処理

手順 1 で作成した対訳単語辞書には，明らかに不適切な対訳単語が多数含まれている．そこで，任意の閾値で枝刈り処理を行う．

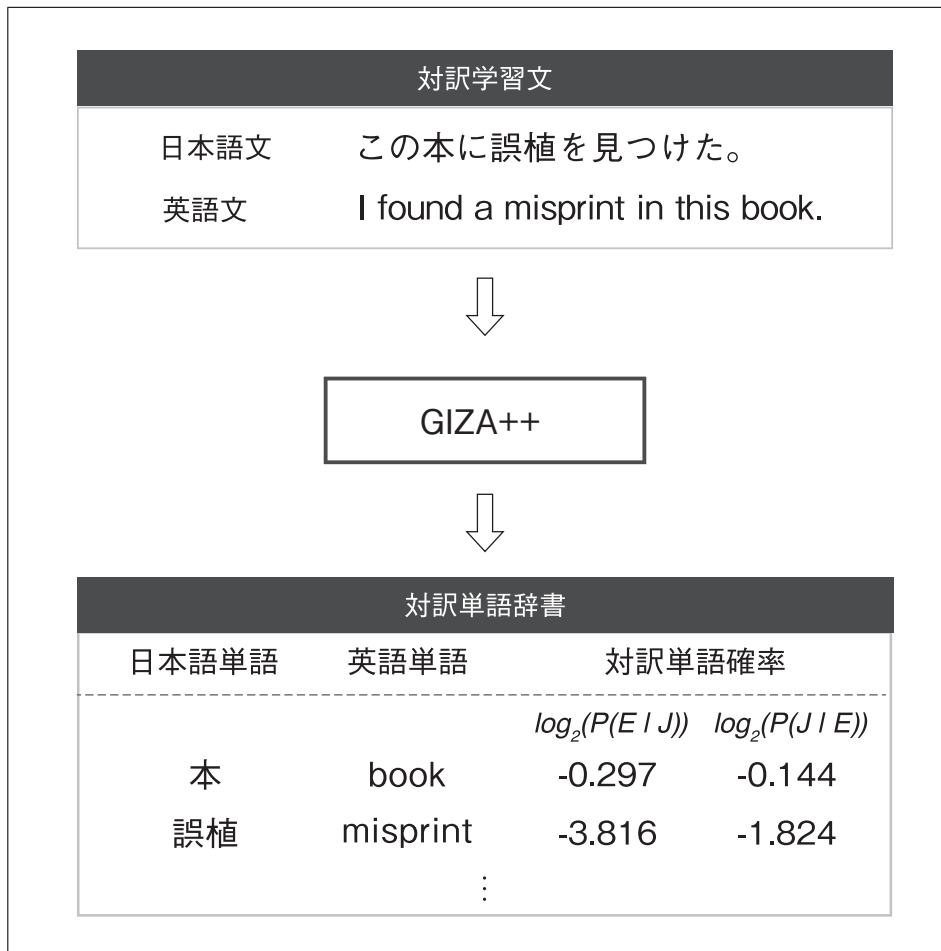


図 2.2: 対訳単語辞書作成の流れ図

## 2.3 IBM 翻訳モデル

IBM 翻訳モデルを以下に示す．以下，川原 [7] の論文を参照して記述している．統計翻訳における単語対応を獲得するための代表的なモデルとして，IBM の Brown らによる仏英翻訳モデル [3] がある．IBM 翻訳モデルは，Model1 から Model5 までの 5 つのモデルから構成されている．各モデルの概要を以下に示す．

Model1 目的言語のある単語が原言語の単語に訳される確率を用いる

Model2 Model1 に加えて，目的言語のある単語に対応する原言語の単語の原言語文中での位置の確率（以下，permutation 確率と呼ぶ）を用いる（絶対位置）

Model3 Model2 に加えて，目的言語のある単語が原言語の何単語に対応するかの確率を用いる

Model4 Model3 の permutation 確率を改良（相対位置）

Model5 Model4 の permutation 確率を更に改良

IBM 翻訳モデルは仏英翻訳を前提としているが，本研究では日英翻訳を扱っているため，日英翻訳を前提に説明する．原言語の日本語文を  $J$ ，目的言語の英語文を  $E$  として定義する．IBM 翻訳モデルにおいて，日本語文  $J$  と英語文  $E$  の翻訳モデル  $P(J|E)$  を計算するため，アライメント  $a$  を用いる．以下に IBM 翻訳モデルの基本的な計算式を示す．

$$P(J|E) = \sum_a P(J, a|E) \quad (2.1)$$

ここで，アライメント  $a$  は， $J$  と  $E$  の単語の対応を意味している．IBM 翻訳モデルにおいて，各日単語に対応する英単語は 1 つであるのに対して，各英単語に対応する日単語は 0 から  $n$  個あると仮定する．また，日単語と適切な英単語が対応しない場合，英語文の先頭に  $e_0$  という空単語があると仮定し，日単語と対応させる．

### 2.3.1 Model1

式 (3) は以下の式に置き換えられる .

$$P(j, a|E) = P(m|E) \prod_{j=1}^m P(a_j|a_1^{j-1}, j_1^{j-1}, m, E) P(j_j|a_1^j, j_1^{j-1}, m, E) \quad (2.2)$$

$m$  は日本語文の文長を示す . また ,  $a_1^{j-1}$  は日本語文の 1 単語目から  $j-1$  単語目までのアライメントである . そして  $j_1^{j-1}$  は日本語文の 1 番目から  $j-1$  番目までの単語を示す . ここで , Model1 では以下を仮定している .

- 日本語文の長さの確率  $\epsilon$  は ,  $m$  と  $E$  に依存しない  
 $\epsilon \equiv P(m|E)$
- アライメントの確率は英語文の長さ  $l$  にのみ依存する  
 $P(a_j|a_1^{j-1}, j_1^{j-1}, m, E) \equiv (l+1)^{-1}$
- 日本語の翻訳確率  $t(j_j|e_{a_j})$  は , 日単語に対応する英単語にのみ依存する  
 $P(j_j|a_1^j, j_1^{j-1}, m, E) \equiv t(j_j|e_{a_j})$

以上の仮定を用いて , 式 (4) は簡略化することができる . 以下に式を示す .

$$P(J, a|E) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^m t(j_j|e_{a_j}) \quad (2.3)$$

$$P(J|E) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^l \cdots \sum_{a_m=0}^l \prod_{j=1}^m t(j_j|e_{a_j}) \quad (2.4)$$

$$= \frac{\epsilon}{(l+1)^m} \prod_{j=1}^m \sum_{i=0}^l t(j_j|e_i) \quad (2.5)$$

Model1 において , 翻訳確率  $t(j|e)$  の初期値が 0 でない場合 , EM アルゴリズムを用いて最適解を推定する . EM アルゴリズムの手順を以下に示す .

手順 1  $t(j|e)$  に初期値を設定する .

手順 2 日本語と英語の対訳文  $(J^{(s)}, E^{(s)})(1 \leq s \leq S)$  において , 日単語  $j$  と英単語  $e$  が対応付けられる回数の期待値を求める . ここで  $\delta(j, j_j)$  は日本語文  $J$  において日単語  $j$  が出現する回数を表す . そして  $\delta(e, e_i)$  は英語文  $E$  において英単語  $e$  が出現する回数を表す .

$$c(j|e; J, E) = \frac{t(j|e)}{t(j|e_0) + \cdots + t(j|e_l)} \sum_{j=1}^m \delta(j, j_j) \sum_{i=0}^l \delta(e, e_i) \quad (2.6)$$

手順3 英語文  $E^{(s)}$  において, 1 回以上出現する英単語  $e$  に対して, 翻訳確率  $t(j|e)$  を計算する.

- 定数  $\lambda_e$  を以下の式で計算する

$$\lambda_e = \sum_j \sum_{s=1}^S c(j|e; J^{(s)}, E^{(s)}) \quad (2.7)$$

- 上式で求めた定数  $\lambda_e$  を用いて  $t(j|e)$  を以下の式で再計算する

$$t(j|e) = \lambda_e^{-1} \sum_{s=1}^S c(j|e; J^{(s)}, E^{(s)}) \quad (2.8)$$

$$= \frac{\sum_{s=1}^S c(j|e; J^{(s)}, E^{(s)})}{\sum_j \sum_{s=1}^S c(j|e; J^{(s)}, E^{(s)})} \quad (2.9)$$

手順4  $t(j|e)$  が収束するまで, 手順2 と手順3 を繰り返す.

### 2.3.2 Model2

Model1 において, アライメントの確率は英語文の長さ  $l$  にのみ依存する. そこで Model2 では, 英語文の長さ  $l$  に加え,  $j$  単語目のアライメント  $a_j$ , 日本語文の長さ  $m$  に依存するとし, 以下の式で表す.

$$a(a_j|j, m, l) \equiv P(a_j|a_1^{j-1}, j_1^{j-1}, m, l) \quad (2.10)$$

よって, Model1 の式 (6) は以下のように置き換えられる.

$$P(J|E) = \epsilon \sum_{a_1=0}^l \cdots \sum_{a_m=0}^l \prod_{j=1}^m t(j_j|e_{a_j}) a(a_j|j, m, l) \quad (2.11)$$

$$= \epsilon \prod_{j=1}^m \sum_{i=0}^l t(j_j|e_i) a(i|j, m, l) \quad (2.12)$$

Model2 において, 対訳文中の英単語  $e$  と日単語  $j$  が対応付けされる回数の期待値である  $c(j|e; J^{(s)}, E^{(s)})$  と, 日単語の位置  $j$  と英単語の位置  $i$  が対応付けられる回数の期待値  $c(i|j, m, l; J^{(s)}, E^{(s)})$  が存在する. 以下に, 期待値  $c(j|e; J^{(s)}, E^{(s)})$  と  $c(i|j, m, l; J^{(s)}, E^{(s)})$  を求める式を示す.

$$c(j|e; J^{(s)}, E^{(s)}) = \sum_{j=1}^m \sum_{i=0}^l \frac{t(j|e) a(i|j, m, l) \delta(j, j_j) \delta(e, e_i)}{t(j|e_0) a(0|j, m, l) + \cdots + t(j|e_l) a(l|j, m, l)} \quad (2.13)$$

$$c(i|j, m, l; J^{(s)}, E^{(s)}) = \frac{t(j_j|e_i) a(i|j, m, l)}{t(j_j|e_0) a(0|j, m, l) + \cdots + t(j_j|e_l) a(l|j, m, l)} \quad (2.14)$$

Model2 においても、最適解を推定するために EM アルゴリズムを用いる。しかし、計算によって複数の極大値が算出され、最適解が得られない場合が存在する。Model2 の特殊な場合に、 $a(i|j, m, l) = (l + 1)^{-1}$  が挙げられるが、これは Model1 として考えることができる。また、最適解が保証されている Model1 で求められた値を初期値として用いることで、最適解を求めることができる。

### 2.3.3 Model3

Model1 および Model2 において、日単語と英単語の対応は 1 対 1 の場合のみを考慮していた。しかし、Model3 では、1 つの単語が複数の単語に対応する場合や、単語の翻訳位置の距離についても考慮する。また、Model3 では単語の位置を絶対位置として考えている。Model3 では以下のパラメータを用いる。

- $P(j|e)$   
英単語  $e$  が日単語  $j$  に翻訳される確率
- $n(\phi|e)$   
英単語  $e$  が  $\phi$  個の日単語と対応する確率
- $d(j|i, m, l)$   
英語文の長さ  $l$ 、日本語文の長さ  $m$  のとき、 $i$  番目の英単語  $e_i$  が  $j$  番目の日単語  $j_j$  に翻訳される確率

さらに、英単語に翻訳されない日本語の単語数を  $\phi_0$  として、そのような単語が発生する確率  $p_0$  を以下の式に表す。

$$P(\phi_0|\phi_1^l, e) = \binom{\phi_1 + \dots + \phi_l}{\phi_0} p_0^{\phi_1 + \dots + \phi_l - \phi_0} p_1^{\phi_0} \quad (2.15)$$

したがって、Model3 は以下の式によって表される。

$$P(j|e) = \sum_{a_1=0}^l \dots \sum_{a_m=0}^l P(j, a|e) \quad (2.16)$$

$$= \sum_{a_1=0}^l \dots \sum_{a_m=0}^l \binom{m - \phi_0}{\phi_0} p_0^{m - 2\phi_0} p_1^{\phi_0} \prod_{i=1}^l \phi_i! n(\phi_i|e_i) \times \prod_{j=1}^m t(j|e_{a_j}) d(j|a_j, m, l) \quad (2.17)$$



Model3 では、全ての単語対応を考慮して計算するため、計算量が膨大となる。そのため、期待値は近似によって求められる。

### 2.3.4 Model4

Model3 と Model4 の違いは、単語の位置の考慮の仕方である。Model3 において、単語の位置は絶対位置で考慮していた。それに対して、Model4 では単語の位置を相対位置で考慮する。また、各単語ごとの位置も考慮している。Model4 では、単語位置の歪みの確率である  $d(j|i, m, l)$  を以下の2通りで考慮する。

- 英単語に対応する日単語が1以上あるときに、その中で最も文頭に近い場合

$$P(\Pi_{[i]1} = j | \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_1(j - \odot_{i-1} | \mathcal{A}(e_{[i-1]}), \mathcal{B}(j_j)) \quad (2.18)$$

- それ以外の場合

$$P(\Pi_{[i]k} = j | \pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_{>1}(j - \pi_{[i]k-1} | \mathcal{B}(j_j)) \quad (2.19)$$

### 2.3.5 Model5

Model4 では、単語の位置に関して直前の単語のみを考慮している。そのため、複数の単語が同じ位置に生じたり、単語が存在しない位置に生成されるという問題がある。Model5 では、この問題を避けるために、単語を空白部分に配置するように制約が施されている。

## 2.4 HMM

単語アライメントの問題について，Stephan ら [4] による 1 次の隠れマルコフモデル（以下、HMM）がある．以下，Stephan らの論文を参照して記述している．HMM の重要な要素は，アライメントの確率を単語アライメントの絶対位置ではなく，相対位置に依存させることである．つまり，インデックス自体ではなく，単語の位置のインデックスの違いを考慮する．

HMM は仏英翻訳を前提としているが，本研究では日英翻訳を扱っているため，日英翻訳を前提に説明する．日本語の文字列  $J_1^J = J_i \dots j_j \dots J_J$  が与えられる．これは英語の文字列  $e_1^I = e_1 \dots e_i \dots e_I$  に変換される．考えられる全ての英語の文字列の中から，ベイズの決定規則によって与えられる最も高い確率を持つものを選択する．

$$\hat{e}_1^I = \arg \max_{e_1^I} \{Pr(e_1^I | j_1^J)\} \quad (2.20)$$

$$= \arg \max_{e_1^I} \{Pr(e_1^I) \cdot Pr(j_1^J | e_1^I)\} \quad (2.21)$$

$Pr(e_1^I)$  は目的言語の言語モデルであるが， $Pr(j_1^J | e_1^I)$  は，文字列翻訳モデルである． $\arg \max$  操作は，検索問題を示す．文字列翻訳確率  $Pr(j_1^J | e_1^I)$  をモデル化するために，確率依存性に構造を導入する．

ここでは，混合ベースのアライメントモデルについて説明する．このモデルは，後で提示する HMM の基本アライメントの参照として使用する．このモデルは，各単語  $j$  の確率について， $j$  の結合確率を積に分解したものに基いている．

$$Pr(j_1^J | e_1^I) = p(J|I) \cdot \prod_{j=1}^J p(j_j | e_1^I) \quad (2.22)$$

$p(J|I)$  は，正規化のための文の長さの確率である．次のステップは，日本語の単語  $j_j$  と各英語の単語  $e_i, i = 1, \dots, I$  との間の一種のペアワイズ相互作用である．これらの依存関係は，混合分布から取り込まれる．

$$Pr(j_1^J | e_1^I) = \sum_{i=1}^I p(i, j_j | e_1^I) \quad (2.23)$$

$$= \sum_{i=1}^I p(i|j, I) \cdot p(j_j | e_i) \quad (2.24)$$

全てをまとめると，以下のような混合ベースモデルがある．

$$Pr(j_1^J | e_1^I) = p(J|I) \cdot \prod_{j=1}^J \sum_{i=1}^I [p(i|j, I) \cdot p(j_j | e_i)] \quad (2.25)$$

- 文の長さの確率： $p(J|I)$
- 混合アライメント確率： $p(i|j, I)$
- 翻訳確率： $p(j|e)$

一様アライメント確率を以下に仮定する．

$$p(i|j, I) = \frac{1}{I} \quad (2.26)$$

このモデルは Brown らが提案した IBM Model1 と呼ばれる [3]．翻訳確率  $p(j|e)$  を学習するために， $[j_1^{j_s}; e_1^{e_s}]$ ,  $s = 1, \dots, S$  からなる対訳コーパスを使用する．最大尤度基準を使用して，以下の反復方程式を得る．

$$\hat{p}(j|e) = \frac{A(j, e)}{\sum_{j'} A(j', e)} \quad (2.27)$$

$$A(j|e) = \sum_s \left( \frac{p(j, e)}{\sum_{i=1}^{I_s} p(j|e_{is})} \sum_{j=1}^{J_s} \delta(j, j_{js}) \sum_{i=1}^{I_s} \delta(e, e_{is}) \right) \quad (2.28)$$

一様アライメント確率の場合，最適値は1つしかないため，EM アルゴリズムが常に大域的最適値を見つけることができる．一様でないアライメント確率を有する混合アライメントモデル（以下，IBM Model2）は，アライメントパラメータ  $p(i|j, I)$  が使用される．したがって，アライメント確率の特定のモデルが使用される．

$$\hat{p}(i|j, I) = \frac{r(i - j \frac{I}{J})}{\sum_{i'=1}^I r(i' - j \frac{I}{J})} \quad (2.29)$$

このモデルを学習するために，最大近似法において最尤基準を使用する．すなわち，尤基準は全アライメントの集合ではなく，最も可能性の高いアライメントのみをカバーする．

$$Pr(j_1^J | e_1^I) \cong \prod_{j=1}^J \max_i [p(i|j, I) \cdot p(j|e_i)] \quad (2.30)$$

学習では，この基準は一連の反復に相当し，各反復は2つのステップで構成されている．

- 位置アライメント：モデルパラメータを大きくして，最も可能性の高い位置アライメントを決定する．
- パラメータ推定：全ての文対に対するアライメント経路に沿って進むと仮定して，モデルパラメータの最大尤度推定を実行する．モデル分布については，これらの相対度数になる．

混合モデルの性質上，隣接する単語位置間の相互作用はない．したがって，各位置  $j$  に対する最適位置  $i$  は，隣接位置とは無関係に決定できる．

次に HMM モデルについて説明する．ここでは，HMM 単語単位の代入を記述するために，位置  $j$  の単語  $j_j$  を位置  $j \rightarrow a_j$  の  $e_i$  に割り当てる写像  $i = a_j$  を導入する．位置  $j$  に対するアライメント  $a_j$  の確率は，前のアライメント  $a_{j-1}$  に依存する．

$$p(a_j|a_{j-1}, I) \quad (2.31)$$

正規化のために，英語の文の全長  $I$  に対する条件付けを含めた．したがって，問題の定式化は，いわゆる隠れマルコフモデルが使用されてきた音声認識における時間整合問題と類似している [5]．同じ基本原則を使用して，文対  $[j_1^J; e_1^I]$  に対して，「隠れた」アライメント  $a_1^J := a_1 \dots a_j \dots a_J$  を導入することで書き換えることができる．

$$Pr(j_1^J|e_1^I) = \sum a_1^J Pr(j_1^J, a_1^J|e_1^I) \quad (2.32)$$

$$= \sum_{a_1^J} \prod_{j=1}^J Pr(j_1^{J-1}, a_1^{J-1}, e_1^I) \quad (2.33)$$

これまでのところアプローチの基本的な制限はない．ここで，線形  $a_j$  のみへの一次依存性を仮定する．

$$Pr(j_1^{J-1}, a_1^{J-1}, e_1^I) \quad (2.34)$$

$$= p(j_j, a_j|a_{j-1}, e_1^I) \quad (2.35)$$

$$= p(a_j|a_{j-1}, I) \cdot p(j_j|e_{a_j}) \quad (2.36)$$

ここでさらに，翻訳確率は  $a_j$  のみに依存し， $a_{j-1}$  には依存しないと仮定した．まとめると，以下のような HMM ベースモデルがある．

$$Pr(j_1^J|e_1^I) = \sum a_1^J \prod_{j=1}^J [p(a_j|a_{j-1}, I) \cdot p(j_j|e_{a_j})] \quad (2.37)$$

- HMM アライメント確率：  $p(i|i', I)$  または  $p(a_j|a_{j-1}, I)$
- 翻訳確率：  $p(j|e)$

さらに，HMM アライメント確率：  $p(i|i', I)$  はジャンプ幅  $(i - i')$  にのみ依存すると仮定する．負でないパラメータの集合  $\{s(i - i')\}$  を使って，HMM アライメント確率を以下の形式で書くことができる．

$$p(i|i', I) = \frac{s(i - i')}{\sum_{I=1}^I s(I - i')} \quad (2.38)$$

この形式は、各単語位置  $i', i' = 1, \dots, I$  に対して、HMM アライメント確率が正規化制約を満たすことを保証する。混合モデルは一次 HMM とは対照的に、ゼロ次モデルとして解釈できる。IBM Model2 と同様に、ここでも最大近似を使用する。

$$Pr(j_1^J | e_1^I) \cong \max_{a_1^J} \prod_{j=1}^J [p(a_j | a_{j-1}, I) \cdot p(j_j | e_{a_j})] \quad (2.39)$$

この場合、最適アライメントを見出す作業は、混合モデル (IBM Model2) の場合よりも複雑である。したがって、以下の典型的な再帰式がある動的計画法に頼る必要がある。

$$Q(i, j) = p(j_j | e_i) \max_{i'=1, \dots, I} [p(i | i', I) \cdot Q(i', J - 1)] \quad (2.40)$$

ここで、 $Q(i, j)$  は、音声認識のタイムアライメントのような一種の部分確率である。

## 2.5 GIZA++

GIZA++の概要を以下に示す。以下、川原 [7] の論文を参照して記述している。GIZA++ [6] とは、日英方向と英日方向の対訳文から最尤な単語対応を得るための計算を行うツールである。IBM 翻訳モデルの Model1 から Model5 に基づいて、単語の対応関係の確率値を計算する。GIZA++を用いた場合、以下の2つのファイルが出力される。

1. **T TABLE (Translation Table)** T TABLE は、Model1 から Model3 により作成された翻訳確率  $P(f|e)$  のデータである。 $f$  は翻訳する言語で、 $e$  は目的言語である。T TABLE は各行が、目的言語の単語 ID( $e_id$ )、翻訳する言語の単語 ID( $f_id$ )、翻訳する言語の単語から目的言語の単語へ翻訳する確率 ( $P(f_id|e_id)$ ) で構成される。
2. **N TABLE (Fertility Table)** N TABLE は、目的言語の単語における繁殖数を表したデータである。N TABLE は各行が、目的言語の単語 ID( $e_id$ )、繁殖数が 0 である確率 ( $p_0$ )、繁殖数が 1 である確率 ( $p_1$ )、...、繁殖数が  $n$  である確率 ( $p_n$ ) で構成される。

## 2.6 中村らの研究 [2]

### 2.6.1 概要

中村ら [2] は、対訳単語辞書の精度調査を行った。対訳学習文と GIZA++ を利用して対訳単語を作成し、全対訳単語から枝刈りを行い、対訳単語辞書を作成している。作成した対訳単語辞書において、日本語単語と英語単語の適切な対応の数を調査した。調査結果より、ひらがな 1 文字・数字・記号において不適切な対訳単語が多いと報告されている。

### 2.6.2 問題点

中村らの研究の問題点を以下に示す。

- (a) GIZA++ の対訳単語確率の計算において、IBM Model1 のみの調査であり、IBM Model2 ~ Model5 での調査は行われていない。
- (b) 対訳単語辞書に含まれる全ての対訳単語を調査対象としており、未知語処理を対象とした調査ではない。

## 第3章 調査手法

### 3.1 調査手法の概要

本研究では，IBM Model1 ~ Model5 と HMM において頻度 1 単語の精度を調査する．具体的には，対訳学習文と GIZA++ の対訳単語確率（IBM Model1 ~ Model5 と HMM）を用いて対訳単語を作成する．作成した対訳単語から枝刈りを行い，対訳単語辞書を作成している．そして対訳単語辞書に含まれる頻度 1 単語の精度を調査する．

### 3.2 研究の目的

中村ら [2] は，GIZA++ の対訳単語確率の計算において，IBM Model1 のみの調査であり，IBM Model2 ~ Model5 での調査は行われていない．また，対訳単語確率の計算において，IBM Model1 のみの調査であり，IBM Model2 ~ Model5 での調査は行われていない．

そこで本研究では，GIZA++ の対訳単語確率の計算において，IBM Model1 ~ Model5 と HMM を用いて対訳単語辞書を作成する．また未知語の多くが頻度 1 単語であることから頻度 1 単語を調査する．なお，本論文の頻度 1 単語は，対訳学習文中の日本語単語と英語単語が共に出現する頻度が 1 回の単語を指す．



## 第4章 実験

### 4.1 実験データ

本研究には、単文のみを用いて調査を行う。本研究で用いる単文コーパスの例を示す。

表 4.1.1: 単文コーパスの例

日本語文	彼は窓のそばにじっと立っていた。
英語文	He was standing still by the window .
日本語文	緑色は彼によく似合う。
英語文	He looks good in green .
日本語文	老人は足から衰える。
英語文	Old people grow weak in the legs first .

調査は、電子辞書などの例文より抽出した単文コーパス [8] の対訳学習文 160,000 文を用いる。コーパスの前処理として、各コーパスの日本語文に対して、“MeCab[9]”を用いて形態素解析を行う。また、英語文に対して“tokenizer.perl[10]”を用いて分かち書きを行う。

### 4.2 調査条件

GIZA++の対訳単語確率の計算は、IBM Model1 ~ Model5 の5つのモデル (Model1 ~ Model3 は、HMM と組み合わせる) で行う。各モデルのGIZA++のパラメータを以下に示す。なお、パラメータの左辺はモデルを表し、右辺の数値は学習回数を表す。

**Model1** m1=4, m2=0, mh=4, m3=0, m4=0, m5=0

**Model2** m1=0, m2=4, mh=4, m3=0, m4=0, m5=0

**Model3** m1=0, m2=0, mh=4, m3=4, m4=0, m5=0

**Model4** m1=0, m2=0, mh=0, m3=0, m4=4, m5=0

**Model5** m1=0, m2=0, mh=0, m3=0, m4=0, m5=4

### 4.3 枝刈り条件

対訳単語辞書作成の際に，誤った対訳単語が大量に生成する．そこで，対訳単語辞書を作成する際に，以下の条件で枝刈りを行う．

- GIZA++の対訳単語確率が，日本語・英語ともに  $\log_2(0.01)$  より高い単語
- GIZA++の対訳単語確率を用いて作成した対訳単語の順位が，日本語・英語それぞれ8位以内の単語

### 4.4 評価方法

評価は全て，評価者1名で人手評価を行う．具体的には，対訳学習文とGIZA++を用いて作成した対訳単語辞書から頻度1単語をランダムで100単語取り出して評価する．評価基準を表4.4.1に示す．

表 4.4.1: 評価基準

○：適切な対訳単語

×：不適切な対訳単語

## 第5章 実験結果

### 5.1 評価結果

IBM Model1 ~ 5 の頻度 1 単語の精度を調査した．評価結果を表 5.1.1 に示す．表中の全単語数は，作成した対訳単語辞書の全単語の数である．

表 5.1.1: 評価結果

モデル	×	頻度 1 単語数	(全単語数)
Model1	<b>37 63</b>	52,271	86,857
Model2	32 68	51,757	86,401
Model3	33 67	46,083	76,679
Model4	32 68	48,575	82,838
Model5	34 64	47,980	82,136

表 5.1.1 より，頻度 1 単語は全てのモデルで全単語数の約 60 %であった．つまり，頻度 1 単語が非常に多いことが分かる．頻度 1 単語の精度は，Model1 がやや高かったが，どのモデルも大きな差はなかった．

## 5.2 頻度1単語の種類別の評価

本節では，5.1節で調査した頻度1単語をひらがな，カタカナ，漢字と分けて評価した．対訳単語の種類別の評価を表5.2.1に示す．

表 5.2.1: 頻度1単語の種類別の評価結果

モデル	ひらがな	カタカナ	漢字
Model1	0.200 (3/15)	<b>0.615 (8/13)</b>	<b>0.366 (26/71)</b>
Model2	0.200 (3/15)	0.230 (3/13)	0.352 (25/71)
Model3	<b>0.428 (6/14)</b>	0.466 (7/15)	0.275 (19/69)
Model4	0.312 (5/16)	0.200 (2/10)	0.328 (24/73)
Model5	0.312 (5/16)	0.300 (3/10)	0.356 (26/73)

表 5.2.1 を見ると，Model1 と Model2 ではひらがなの対訳単語の精度が漢字の対訳単語の精度より低かった．一方，Model3 ではひらがなの対訳単語の精度が漢字の対訳単語の精度より高かった．また，Model4 と Model5 ではひらがなの対訳単語と漢字の対訳単語の精度にほとんど差はなかった．

### 5.3 作成した対訳単語辞書に含まれる例

作成した対訳単語辞書に含まれる例を表 5.3.1 ~ 表 5.3.8 に示す．また，表中の各項目の概要を以下に示す．

- $\log_2(P(E|J))$  : 日本語単語が英語単語に訳される GIZA++ の対訳単語確率
- $\log_2(P(J|E))$  : 英語単語が日本語単語に訳される GIZA++ の対訳単語確率
- 日順 : 対訳単語辞書の日本語単語を  $\log_2(P(E|J))$  の順に並べた順位
- 英順 : 対訳単語辞書の英語単語を  $\log_2(P(J|E))$  の順に並べた順位
- 日頻 : 対訳学習文中に出現する日本語単語の単語数
- 英頻 : 対訳学習文中に出現する英語単語の単語数
- 共頻 : 対訳学習文中に日本語単語と英語単語がともに出現する単語数

表 5.3.1: 作成した対訳単語辞書に含まれる日本語単語「連合」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	連合	×	-5.375	7	44	1
	Therefore		-5.021	4	21	
Model1	連合		-5.400	8	44	1
	Unions		-4.458	5	1	
Model2	連合	×	-5.395	7	44	1
	Therefore		-4.931	3	21	
Model3	連合	×	-5.629	8	44	1
	Therefore		-5.311	7	21	
Model4	連合		-5.790	5	44	1
	combination		-5.057	6	21	
Model5	連合		-5.878	5	44	1
	combination		-5.088	5	21	

表 5.3.1 において，日本語単語「連合」に対して，Model1，Model4，Model5 では，英語単語「Unions」や「combination」という適切な対訳単語が生成された．しかし，Model1 ~ Model3 では，英語単語「Therefore」という不適切な対訳単語が生成された．

表 5.3.2: 作成した対訳単語辞書に含まれる日本語単語「備品」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	備品	×	-3.589	2	5	1
	requisition		-3.176	1	2	
Model2	備品	×	-3.599	2	5	1
	requisition		-3.158	2	2	
Model3	備品		-5.317	8	5	1
	fixtures		-2.450	2	1	
Model3	備品	×	-3.735	2	5	1
	requisition		-3.347	2	2	
Model4	備品	×	-4.250	3	5	1
	benches		-3.169	1	1	
Model4	備品		-4.250	4	5	1
	fixtures		-2.493	2	1	
Model4	備品	×	-4.250	2	5	1
	requisition		-3.315	2	2	
Model5	備品	×	-4.181	2	5	1
	benches		-3.004	1	1	
Model5	備品		-4.182	4	5	1
	fixtures		-2.487	2	1	
Model5	備品	×	-4.181	3	5	1
	requisition		-3.301	2	2	

表 5.3.2 において，Model3 ~ Model5 では，日本語単語「備品」に対して，英語単語「fixtures」という適切な対訳単語が生成された．しかし，「requisition」や「benches」など，不適切な対訳単語が生成された．

表 5.3.3: 作成した対訳単語辞書に含まれる日本語単語「長寿」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	長寿		-3.202	2	4	1
	longevity		-3.317	3	4	
Model1	長寿	×	-3.694	4	4	1
	ninety		-5.108	5	14	
Model2	長寿	×	-6.537	6	4	1
	centenarians		-1.963	1	1	
Model2	長寿		-4.827	3	4	1
	longevity		-4.688	2	4	
Model2	長寿	×	-3.090	2	4	1
	ninety		-5.114	5	14	
Model3	長寿	×	-3.718	2	4	1
	ninety		-5.173	5	14	
Model4	長寿	×	-3.168	2	4	1
	Long		-5.648	6	27	
Model5	長寿	×	-3.193	3	4	1
	ninety		-5.307	5	14	
Model5	長寿	×	-3.150	2	4	1
	Long		-5.725	7	27	
Model5	長寿	×	-3.304	3	4	1
	ninety		-5.271	4	14	

表 5.3.3 において, Model1 と Model2 では, 日本語単語「長寿」に対して, 英語単語「longevity」という適切な対訳単語が生成された. しかし, どのモデルでも「ninety」や「Long」など, 不適切な対訳単語が生成された.

表 5.3.4: 作成した対訳単語辞書に含まれる日本語単語「ふた」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	ふた	×	-5.930	6	56	1
	Fit		-6.304	3	3	
Model2	ふた	×	-5.920	7	56	1
	Fit		-4.010	3	3	
Model3	ふた	×	-6.074	7	56	1
	Fit		-4.467	4	3	
Model4	ふた	×	-6.157	6	56	1
	phial		-2.550	1	1	
Model5	ふた	×	-6.019	6	56	1
	phial		-2.546	2	1	

表 5.3.4 において，どのモデルでも，日本語単語「ふた」に対して，適切な対訳単語が生成されなかった．この例のようなひらがなの対訳単語は，全体的に精度が低かった．



表 5.3.5: 作成した対訳単語辞書に含まれる日本語単語「アグネス」の全ての例

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	アグネス		-1.013	1	1	1
	Agnes		-0.604	1	1	
Model2	アグネス		-1.015	1	1	1
	Agnes		-0.520	1	1	
Model3	アグネス		-1.476	1	1	1
	Agnes		-1.196	1	1	
Model3	アグネス	×	-2.462	2	1	1
	angelic		-5.845	2	1	
Model4	アグネス	×	-2.318	1	1	1
	angelic		-2.084	1	1	
Model5	アグネス	×	-2.291	1	1	1
	angelic		-2.317	1	1	

表 5.3.5 において，Model1 ~ Model3 では，日本語単語「アグネス」に対して，英語単語「Agnes」という適切な対訳単語が生成された．しかし，Model3 ~ Model5 では「angelic」という不適切な対訳単語が生成された．この例のようなカタカナの対訳単語は，地名や人名などの固有名詞が多かったが，誤った対訳単語を生成する単語もあった．

表 5.3.6: 作成した対訳単語辞書に含まれる日本語単語「1932」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	1932		-1.234	1	1	1
Model1	1932		-1.604	1	1	1
Model1	1932	×	-5.481	2	1	1
Model1	Tsuyoshi	×	-2.860	3	1	1
Model2	1932		-1.107	1	1	1
Model2	1932		-1.523	1	1	1
Model3	1932		-1.896	1	1	1
Model3	1932		-2.376	1	1	1
Model3	1932	×	-5.154	4	1	1
Model3	Inukai	×	-3.411	3	1	1
Model3	1932	×	-3.186	3	1	1
Model3	Tsuyoshi	×	-2.718	3	1	1
Model4	1932		-2.535	2	1	1
Model4	1932		-0.992	1	1	1
Model4	1932	×	-2.733	3	1	1
Model4	Inukai	×	-5.904	1	1	1
Model5	1932		-2.204	1	1	1
Model5	1932		-1.011	1	1	1

表 5.3.6 において、どのモデルも日本語単語「1932」に対して、英語単語「1932」という適切な対訳単語が生成された。しかし、Model1, Model3, Model4 において「Inukai」や「Tsuyoshi」などの不適切な対訳単語が生成された。この例のような数字の対訳単語は、全体的に精度は高かったが、不適切な対訳単語も生成された。

表 5.3.7: 作成した対訳単語辞書に含まれる日本語単語「ISO」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	ISO	×	-2.373	1	1	1
	14001		-5.186	4	2	
Model1	ISO		-2.373	2	1	1
	ISO		-3.400	2	2	
Mode2	ISO	×	-2.345	1	1	1
	14001		-5.642	4	2	
Model2	ISO		-2.345	2	1	1
	ISO		-3.272	2	2	
Model3	ISO		-2.756	2	1	1
	ISO		-4.223	4	2	
Mode4	ISO	×	-3.158	1	1	1
	14001		-3.574	2	2	
Mode4	ISO	×	-2.864	2	1	1
	14001		-3.582	1	2	

表 5.3.7 において，Model1 ~ Model3 において，英語単語「ISO」という適切な対訳単語が生成された．しかし，Model1，Model2，Model4，Model5 において，「14001」という不適切な対訳単語が生成された．この例のようなアルファベットの対訳単語は，数字と同様に精度が高かったが，不適切な対訳単語も生成された．

表 5.3.8: 作成した対訳単語辞書に含まれる日本語単語「=」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	=	×	-0.559	1	2	1
	Anglo-Saxon		-2.851	2	2	
Model1	=	×	-5.298	2	2	1
	Nprman-French		-2.408	3	2	
Model2	=	×	-0.757	1	2	1
	Anglo-Saxon		-2.862	4	2	
Model3	=	×	-0.723	1	2	1
	Anglo-Saxon		-3.188	1	2	
Model3	=	×	-5.832	3	2	1
	Norman-French		-2.673	3	1	
Model3	=	×	-5.847	4	2	1
	superstratum		-2.850	4	1	
Model4	=	×	-1.596	1	2	1
	Anglo-Saxon		-3.435	1	2	
Model5	=	×	-1.581	1	2	1
	Anglo-Saxon		-3.341	1	2	

表 5.3.8 において，どのモデルも日本語単語「=」に対して，英語単語「Anglo-Saxon」や「Norman-French」などの不適切な対訳単語が生成された．この例のような記号の対訳単語精度は，かなり低かった．

## 第6章 考察

### 6.1 調査対象の変更

#### 6.1.1 作成した対訳単語辞書の全単語

本項では，作成した対訳単語辞書に含まれる全単語を調査対象とする．評価方法は，全単語からランダムで100語取り出して，以下4.4節と同様に評価を行う．

##### 6.1.1.1 評価結果

作成した対訳単語辞書の全単語を調査対象とした評価結果を表6.1.1に示す．

表 6.1.1: 全単語の評価結果

モデル	×		全単語数
Model1	55	45	86,857
Model2	53	47	86,401
Model3	<b>56</b>	<b>44</b>	76,679
Model4	54	46	82,838
Model5	52	48	82,136

表5.1.1と比較すると，全単語の精度は頻度1単語の精度よりも全てのモデルで高かった．全単語数の約60%を頻度1単語が占めることから，頻度2以上の単語の精度が高いと推測できる．

### 6.1.1.2 作成した対訳単語辞書に含まれる例

作成した対訳単語辞書に含まれる日本語単語「尽力」、「本土」、「いか」の例の一部（各モデル日本語順位の上位2単語）を表6.1.2～表6.1.4に示す。

表 6.1.2: 作成した対訳単語辞書に含まれる日本語単語「尽力」の例の一部

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	尽力		-3.354	1	15	3
	services		-4.826	4	100	
Model1	尽力		-3.657	2	15	3
	offices		-5.271	8	78	
Model2	尽力		-3.344	1	15	3
	services		-4.847	4	100	
Model2	尽力		-3.656	2	15	3
	offices		-5.316	8	78	
Model3	尽力		-3.359	1	15	3
	services		-4.766	3	100	
Model3	尽力	×	-5.084	6	15	1
	appreciative		-4.549	5	5	
Model4	尽力		-5.125	1	15	3
	services		-3.715	3	100	
Model4	尽力	×	-5.298	3	15	1
	appreciative		-4.749	4	5	
Model5	尽力		-3.720	1	15	3
	services		-5.141	3	100	
Model5	尽力		-4.305	2	15	3
	offices		-5.874	8	78	

表6.1.2において、日本語単語「尽力」に対して全てのモデルで英語単語「services」が生成された。しかし、Model3とModel4では「appreciative」という不適切な対訳単語も生成された。

表 6.1.3: 作成した対訳単語辞書に含まれる日本語単語「本土」の例の一部

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	本土		-1.376	1	18	14
	mainland		-0.727	1	16	
Model1	本土	×	-5.132	4	18	2
	islands		-5.752	4	50	
Model2	本土		-1.435	1	18	14
	mainland		-0.740	1	16	
Model2	本土	×	-5.238	6	18	1
	Patch		-2.413	2	1	
Model3	本土		-1.584	1	18	14
	mainland		-0.997	1	16	
Model3	本土	×	-5.333	5	18	1
	destinations		-3.987	5	1	
Model4	本土		-2.050	1	18	14
	mainland		-0.694	1	16	
Model4	本土	×	-5.759	2	18	2
	islands		-6.261	5	50	
Model5	本土		-2.129	1	18	14
	mainland		-0.758	1	16	
Model5	本土	×	-5.909	4	18	2
	islands		-6.180	6	50	

表 6.1.3 において，日本語単語「本土」に対して全てのモデルで英語単語「mainland」が生成された．しかし，どのモデルも「Patch」や「islands」などの不適切な対訳単語も生成された．

表 6.1.4: 作成した対訳単語辞書に含まれる日本語単語「いか」の例の一部

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	いか didn't	×	-4.565 -5.942	4 5	79 321	9
Model1	いか poorly	×	-4.820 -3.377	5 1	79 22	3
Model2	いか didn't	×	-4.545 -5.939	4 5	79 321	9
Model2	いか poorly	×	-4.921 -3.405	6 1	79 22	3
Model3	いか didn't	×	-3.864 -5.925	1 5	79 321	9
Model3	いか poorly	×	-4.823 -3.524	7 1	79 22	3
Model4	いか didn't	×	-3.360 -6.527	1 3	79 321	9
Model4	いか poorly	×	-4.724 -3.752	5 1	79 22	3
Model5	いか poorly	×	-4.638 -3.947	5 1	79 22	3
Model5	いか pan	×	-5.226 -4.543	8 4	79 30	2

表 6.1.4 において，日本語単語「いか」に対して適切な対訳単語を生成したモデルはなかった．この例は全て頻度 2 以上の単語であったが，精度が低かった．



## 6.1.2 頻度 2 以上の単語

本項では，作成した対訳単語辞書に含まれる頻度 2 以上の単語を調査対象とする．評価方法は，頻度 2 以上の単語からランダムで 100 語取り出して，以下 4.4 節と同様に評価を行う．

### 6.1.2.1 評価結果

頻度 2 以上の単語の評価結果を表 6.1.5 に示す．

表 6.1.5: 頻度 2 以上の単語の評価結果

モデル	×		全単語数
Model1	81	19	34,586
Model2	76	24	34,644
Model3	78	22	30,596
Model4	76	24	34,263
Model5	80	20	34,156

表 6.1.5 より，頻度 2 以上の単語の精度は，全てのモデルで約 80 %であった．頻度 1 単語の精度と比較して，非常に高かった．

### 6.1.2.2 作成した対訳単語辞書に含まれる例

作成した対訳単語辞書に含まれる日本語単語「いくぶん」、「イナゴ」、「流入」の例の一部（各モデル日本語順位の上位2単語）を表6.1.6～表6.1.8に示す。

表 6.1.6: 作成した対訳単語辞書に含まれる日本語単語「いくぶん」の例の一部

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	いくぶん		-4.430	3	43	3
	slightly		-6.226	6	102	
Model1	いくぶん		-1.915	1	43	14
	somewhat		-2.443	2	63	
Model2	いくぶん		-4.446	3	43	3
	slightly		-6.279	6	102	
Model2	いくぶん		-1.907	1	43	14
	somewhat		-2.443	2	63	
Model3	いくぶん		-2.185	1	43	14
	somewhat		-2.764	2	63	
Model4	いくぶん	×	-5.772	3	43	4
	rather		-6.094	8	180	
Model4	いくぶん		-2.089	1	43	14
	somewhat		-2.654	1	63	
Model5	いくぶん		-4.819	2	43	3
	slightly		-6.565	7	102	
Model2	いくぶん		-2.125	1	43	14
	somewhat		-2.755	1	63	

表6.1.6において、日本語単語「いくぶん」に対して全てのモデルで英語単語「somewhat」という適切な対訳単語が生成された。ひらがなの対訳単語は、頻度1では精度が低かったが、頻度2以上の単語の精度は高い対訳単語が多かった。

表 6.1.7: 作成した対訳単語辞書に含まれる日本語単語「イナゴ」の例の一部

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	イナゴ		-2.237	1	5	4
	locusts		-2.086	1	8	
Model1	イナゴ	×	-3.469	2	5	3
	swarm		-4.736	7	15	
Model2	イナゴ		-2.305	1	5	4
	locusts		-2.296	2	8	
Model2	イナゴ	×	-3.415	2	5	3
	swarm		-4.155	6	15	
Model3	イナゴ		-2.554	1	5	4
	locusts		-2.820	2	8	
Model3	イナゴ	×	-3.168	5	5	3
	swarm		-4.238	2	15	
Model4	イナゴ		-2.649	1	5	4
	locusts		-2.394	2	8	
Model5	イナゴ		-2.374	1	5	4
	locusts		-2.378	2	8	

表 6.1.7 において，日本語単語「イナゴ」に対して全てのモデルで英語単語「locusts」が生成された．しかし，Model1～Model3 では「swarm」という不適切な対訳単語も生成された．

表 6.1.8: 作成した対訳単語辞書に含まれる日本語単語「流入」の例の一部

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	流入	×	-4.008	2	14	2
	rivers		-5.635	6	26	
Model2	流入	×	-4.006	2	14	2
	rivers		-5.607	6	26	
Model3	流入	×	-3.928	2	14	2
	rivers		-5.881	8	26	
Model4	流入	×	-5.298	3	14	2
	pulmonary		-2.360	1	3	
Model4	流入	×	-4.366	1	14	2
	rivers		-6.130	3	26	
Model5	流入	×	-5.329	4	14	2
	pulmonary		-2.145	1	3	
Model5	流入	×	-4.354	2	14	2
	rivers		-6.108	3	26	

表 6.1.8 において，日本語単語「流入」に対して適切な対訳単語を生成したモデルはなかった．頻度 2 以上の単語でも，精度が低い単語があることが分かる．

## 6.2 学習回数の変更

本節では, IBM Model1 を用いて学習回数を 2,8,16 回と変更して調査を行う. GIZA++ のパラメータを以下に示す.

2 回  $m_1=2, m_2=0, m_h=2, m_3=0, m_4=0, m_5=0$

4 回  $m_1=4, m_2=0, m_h=4, m_3=0, m_4=0, m_5=0$

8 回  $m_1=8, m_2=0, m_h=8, m_3=0, m_4=0, m_5=0$

16 回  $m_1=16, m_2=0, m_h=16, m_3=0, m_4=0, m_5=0$

### 6.2.1 評価結果

学習回数が 2,4,8,16 回の評価結果を表 6.2.1 に示す.

表 6.2.1: 学習回数が 2,4,8,16 回の評価結果

学習回数	×	頻度 1 単語数	(全単語数)
2	31 69	48,050	79,797
4	<b>37 63</b>	52,271	86,857
8	31 69	52,213	87,720
16	33 67	51,896	87,452

表 6.2.1 より, 頻度 1 単語の精度は学習回数が 4 回の時がやや高かったが, どの学習回数でも大きな差はなかった.

## 6.2.2 頻度 1 単語の種類別の評価

本項では，各学習回数で翻訳精度の特徴を調べるために，6.2.1 項で調査した頻度 1 単語ををひらがな，カタカナ，漢字と分けて評価した．頻度 1 単語の種類別の評価を表 6.2.2 に示す．

表 6.2.2: 頻度 1 単語の種類別の評価結果

学習回数	ひらがな	カタカナ	漢字
2	0.214 (3/14)	0.400 (6/15)	0.304 (21/69)
4	0.200 (3/15)	<b>0.615 (8/13)</b>	0.366 (26/71)
8	<b>0.250 (4/16)</b>	0.500 (5/10)	0.301 (22/73)
16	0.125 (2/16)	0.400 (4/10)	<b>0.369 (27/73)</b>

表 6.2.2 より，学習回数が 16 回の時は，他の学習回数の時よりひらがなの対訳単語の翻訳精度が低く，漢字の対訳単語の翻訳精度が高かった．この原因は，以下のように考えている．EM 推定においては，学習回数を増やしていくと翻訳候補が一意に絞られやすくなる傾向がある．そのため，一つの意味しか持たない漢字の対訳単語は，複数の意味を持つひらがなの対訳単語より，翻訳精度が高くなる．

### 6.2.3 作成した対訳単語辞書に含まれる例

作成した対訳単語辞書に含まれる日本語単語「かまぼこ」、「スロベニア」、「贈呈」の全ての例を表 6.2.3～表 6.2.5 に示す。

表 6.2.3: 作成した対訳単語辞書に含まれる日本語単語「かまぼこ」の全ての例

学習回数	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
2	かまぼこ		-2.917	1	1	1
	kamaboko		-2.858	3	1	
2	かまぼこ	×	-2.964	2	1	1
	patty		-1.587	1	1	
4	かまぼこ		-2.977	1	1	1
	kamaboko		-6.340	2	1	
4	かまぼこ	×	-2.980	2	1	1
	patty		-1.476	1	1	
8	かまぼこ	×	-2.974	2	1	1
	patty		-1.480	1	1	
16	かまぼこ	×	-2.368	1	1	1
	patty		-1.572	1	1	

表 6.2.3 において、学習回数が 2 回と 4 回の際は、日本語単語「かまぼこ」に対して、英語単語「kamaboko」と「patty」という複数の対訳単語が生成された。しかし学習回数が 8 回、16 回となると適切な対訳単語である「kamaboko」が枝刈りされ、不適切な対訳単語である「patty」のみが生成された。

表 6.2.4: 作成した対訳単語辞書に含まれる日本語単語「スロベニア」の全ての例

学習回数	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
2	スロベニア	×	-2.320	3	1	1
	Elections		-4.163	5	2	
2	スロベニア		-2.320	1	1	1
	Slovenia		-3.882	3	2	
4	スロベニア	×	-2.204	2	1	1
	Elections		-3.548	2	2	
4	スロベニア		-2.204	1	1	1
	Slovenia		-5.116	3	2	
8	スロベニア	×	-2.239	1	1	1
	Elections		-3.283	2	2	
16	スロベニア	×	-2.248	1	1	1
	Elections		-3.207	2	2	

表 6.2.4 において、学習回数が 2 回と 4 回の時は、日本語単語「スロベニア」に対して、英語単語「Elections」と「Slovenia」という複数の対訳単語が生成された。しかし学習回数が 8 回、16 回となると適切な対訳単語である「Slovenia」が枝刈りされ、不適切な対訳単語である「Elections」のみが生成された。



表 6.2.5: 作成した対訳単語辞書に含まれる日本語単語「贈呈」の全ての例

学習回数	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
2	贈呈	×	-4.805	5	8	1
	Okamura		-3.245	1	1	
4	贈呈	×	-5.550	8	8	1
	\$4,000		-3.217	2	1	
4	贈呈	×	-4.591	3	8	1
	Okamura		-3.281	1	1	
8	贈呈	×	-4.609	2	8	1
	Okamura		-3.280	1	1	
8	贈呈		-4.683	5	8	1
	presentation		-4.975	7	16	
16	贈呈	×	-4.600	2	8	1
	Okamura		-3.272	1	1	
16	贈呈		-4.600	4	8	1
	presentation		-5.001	3	16	

表 6.2.5 において，学習回数が 2 回と 4 回の時は，日本語単語「贈呈」に対して，英語単語「Okamura」や「\$4,000」という不適切の対訳単語が生成された．しかし学習回数が 8 回，16 回となると適切な対訳単語である「presentation」が生成された．

## 6.3 順位の枝刈り条件の変更

### 6.3.1 日本語順位 1 位・英語順位 8 位以内

本項では、日本語順位 1 位のみ・英語順位 8 位以内の頻度 1 単語で調査を行う。

#### 6.3.1.1 評価結果

日本語順位 1 位のみ・英語順位 8 位以内の評価結果を表 6.3.1 に示す。

表 6.3.1: 日本語順位 1 位のみ・英語順位 8 位以内の評価結果

モデル	×		頻度 1 単語数	(全単語数)
Model1	45	55	13,834	28,277
Model2	41	59	13,860	28,308
Model3	<b>47</b>	<b>53</b>	10,373	23,170
Model4	35	65	16,426	31,470
Model5	45	55	16,082	30,999

表 5.1.1 と比較すると、全てのモデルで頻度 1 単語の精度が向上した。この結果より、順位が高い頻度 1 単語は、精度が高いといえる。しかし、生成した頻度 1 単語数を比べると、約 25 % に減少した。よって、日本語順位の枝刈り条件を 1 位のみに変更すると頻度 1 単語の精度が向上するが、生成する単語数が大幅に減少する。

### 6.3.1.2 作成した対訳単語辞書に含まれる例

対訳単語辞書に含まれる日本語単語「両生類」、「脛骨」、「異名」の全ての例を表 6.3.2 ~ 表 6.3.4 に示す。

表 6.3.2: 作成した対訳単語辞書に含まれる日本語単語「両生類」の全ての例

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	両生類	×	-2.310	1	2	1
	semiterrestrial		-3.093	1	1	
Model2	両生類		-2.366	1	2	1
	amphibians		-2.799	1	1	
Model3	両生類		-2.799	1	2	1
	amphibians		-2.609	1	1	
Model4	両生類		-3.200	1	2	1
	amphibian		-2.804	1	1	
Model5	両生類	×	-3.147	1	2	1
	semiterrestrial		-1.839	1	1	

表 6.3.2 において，Model2 ~ Model4 では日本語単語「両生類」に対して，英語単語「amphibians」や「amphibian」という適切な対訳単語が日本語順位 1 位となった。しかし，Model1 と Model5 では「semiterrestrial」という不適切な対訳単語が日本語順位 1 位となった。

表 6.3.3: 作成した対訳単語辞書に含まれる日本語単語「脛骨」の全ての例

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	脛骨		-2.238	1	1	1
	tibia		-1.185	1	1	
Model2	脛骨		-2.483	1	1	1
	tibia		-1.130	1	1	
Model3	脛骨	×	-2.673	1	1	1
	osteosarcoma		-5.171	4	1	
Model4	脛骨	×	-3.169	1	1	1
	osteosarcoma		-2.560	1	1	
Model5	脛骨	×	-2.979	1	1	1
	osteosarcoma		-2.529	1	1	

表 6.3.3 において, Model1 と Model2 では日本語単語「脛骨」に対して, 英語単語「tibia」という適切な対訳単語が日本語順位 1 位となった。しかし, Model3 ~ Model5 では, 「osteosarcoma」という不適切な対訳単語が日本語順位 1 位となった。

表 6.3.4: 作成した対訳単語辞書に含まれる日本語単語「異名」の全ての例

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	異名	×	-2.138	1	1	1
	Nezumikozo		-3.003	2	1	
Model2	異名	×	-2.219	1	1	1
	Nezumikozo		-3.136	2	1	
Model3	異名	×	-2.452	1	1	1
	Nezumikozo		-2.881	2	1	
Model4	異名		-2.999	1	1	1
	nicknamed		-2.319	1	1	
Model5	異名		-2.996	1	1	1
	nicknamed		-2.318	1	1	

表 6.3.4 において, Model4 と Model5 では日本語単語「異名」に対して, 英語単語「nicknamed」という適切な対訳単語が日本語順位 1 位となった。しかし, Model1 ~ Model3 では, 「Nezumikozo」という不適切な対訳単語が日本語順位 1 位となった。

## 6.3.2 日本語・英語ともに順位1位

本項では、日本語・英語ともに順位が1位のみの頻度1単語で調査を行う。

### 6.3.2.1 評価結果

日本語・英語ともに順位が1位のみの評価結果を表6.3.5に示す。

表 6.3.5: 日本語・英語ともに順位が1位のみの評価結果

モデル	×	頻度1単語数	(全単語数)
Model1	52 48	5,908	13,065
Model2	52 48	5,952	13,113
Model3	46 54	4,235	10,760
Model4	42 58	8,038	15,510
Model5	42 58	7,907	15,368

順位が日本語・英語ともに1位の単語は、対訳単語確率が最良の単語である。しかし、表6.3.1と比較すると、Model3とModel5では頻度1単語の精度が低下した。よって、順位が日本語・英語ともに1位の頻度1単語が、必ずしも適切ではないことが分かる。

また、表6.1.1と比較しても、精度が低く、頻度1単語数も全単語の10%未満である。つまり、頻度1単語は精度が非常に低いといえる。

### 6.3.2.2 作成した対訳単語辞書に含まれる例

対訳単語辞書に含まれる日本語単語「叱責」、「珍重」、「帝劇」の全ての例を表 6.3.6～表 6.3.8 に示す．なお，生成されなかった単語は「-」で示す．

表 6.3.6: 作成した対訳単語辞書に含まれる日本語単語「叱責」の全ての例

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	叱責		-3.921	1	4	1
	reprehension		-1.832	1	1	
Model2	叱責	×	-3.951	1	4	1
	self-abuse		-2.431	1	1	
Model3	叱責		-4.148	1	4	1
	reprehension		-2.214	1	1	
Model4	叱責	×	-3.784	1	4	1
	self-abuse		-2.982	1	1	
Model5	叱責	×	-4.054	1	4	1
	self-abuse		-2.791	1	1	

表 6.3.6 において，Model1 と Model3 では日本語単語「叱責」に対して，英語単語「reprehension」という適切な対訳単語が日本語・英語ともに順位 1 位となった．しかし，Model2，Model4，Model5 では「self-abuse」という不適切な対訳単語が日本語・英語ともに順位 1 位となった．

表 6.3.7: 作成した対訳単語辞書に含まれる日本語単語「珍重」の全ての例

モデル	日本語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	珍重		-2.947	1	3	1
	rarity		-2.997	1	4	
Model2	珍重		-2.962	1	3	1
	rarity		-3.022	1	4	
Model3	珍重 -	-	-	-	-	-
Model4	珍重	×	-3.001	1	3	1
	collector's		-2.863	1	1	
Model5	珍重		-2.981	1	3	1
	rarity		-3.457	1	4	

表 6.3.7 において, Model1, Model2, Model5 では日本語単語「珍重」に対して, 英語単語「rarity」という適切な対訳単語が日本語・英語ともに順位 1 位となった。しかし, Model4 では「collector's」という不適切な対訳単語が日本語・英語ともに順位 1 位となった。また, Model3 では生成されなかった。

表 6.3.8: 作成した対訳単語辞書に含まれる日本語単語「帝劇」の全ての例

モデル	日語 英語	評価	$\log_2(P(E J))$ $\log_2(P(J E))$	日順 英順	日頻 英頻	共頻
Model1	帝劇	×	-3.025	1	2	1
	hoped-for		-2.234	1	1	
Model2	帝劇	×	-3.047	1	2	1
	haped-for		-2.244	1	1	
Model3	帝劇	×	-3.205	1	2	1
	haped-for		-2.479	1	1	
Model4	帝劇		-3.459	1	2	1
	Teigeki		-1.582	1	1	
Model5	帝劇		-3.582	1	2	1
	Teigeki		-1.514	1	1	

表 6.3.8 において，Model4 と Model5 では日本語単語「帝劇」に対して，英語単語「Teigeki」という適切な対訳単語が日本語・英語ともに順位 1 位となった．しかし，Model1 ~ Model3 では，「hoped-for」という不適切な対訳単語が日本語・英語ともに順位 1 位となった．



## 6.4 未知語の翻訳

本節では，表 5.1.1 で使用した頻度 1 単語の対訳単語辞書を用いて，未知語の翻訳を行う．

### 6.4.1 実験設定

機械翻訳機には，ニューラル機械翻訳 [11] のツールキット OpenNMT [12] を使用する．以下，OpenNMT の設定については，今仁ら [14] の研究を参照して記述している．NMT のツールキットには OpenNMT [12] を用い，モデルは Luong ら [13] により提案された Global Attention を用いる．Encoder，Decoder の LSTM は 2 層とし，ユニット数は 500，単語の分散表現のベクトルサイズは 500 を設定する．ミニバッチサイズは 40 とし，モデルの訓練は最大 32 エポック行う．Optimizer には SGD を使用し，学習率の初期値は 1 とする．また，各エポックごとに得られたモデルを用いてディベロップメント文を翻訳し，BLEU 値 [15] が最高となるモデルを使用する．

### 6.4.2 使用データ

使用データは，ニューラル機械翻訳 [11] を用いて得られた 1,000 文（そのうち未知語を含む文は 213 文）を入力文とする．使用する未知語を含む文の例を示す．

表 6.4.1: 未知語処理 の例

日本語文	その会社は新しいデザイナーを物色中である。
英語文	The company has a new designer in its 物色 .
日本語文	彼は予備校に通っている。
英語文	He goes to 予備校 .
日本語文	我々の権利は蹂躪されるだろう。
英語文	Our rights will be 蹂躪 .

### 6.4.3 自動評価

自動評価は，入力文 1,000 文と各モデルで作成した対訳単語辞書を用いて行う．自動評価には，BLEU[15]，METEOR[16]，RIBES[17]，TER[18]を用いる．

未知語処理した（各モデルの対訳単語辞書を用いる）場合と，未知語処理しない（対訳単語辞書なし）場合の自動評価結果を表 6.4.2 に示す．

表 6.4.2: 自動評価結果

モデル	BLEU	METEOR	RIBES	TER
Model1	0.200	<b>0.486</b>	0.783	<b>0.581</b>
Model2	0.200	<b>0.486</b>	0.783	<b>0.581</b>
Model3	0.200	<b>0.486</b>	0.783	<b>0.581</b>
Model4	0.200	<b>0.486</b>	<b>0.784</b>	<b>0.581</b>
Model5	<b>0.201</b>	<b>0.486</b>	0.783	<b>0.581</b>
なし	0.200	0.485	0.782	0.582

表 6.4.2 より，自動評価は全てのモデルでほとんど差がなかった．この結果の要因の 1 つは，未知語処理が正しく行われた場合でも，文全体の自動評価結果にはあまり影響しないためであると考えている．

## 6.4.4 人手評価

### 6.4.4.1 人手評価方法

使用した入力文において，未知語を含む文が 213 文あった．そのうち，Model1 対訳単語辞書を使用して未知語処理できた文が 86 文あった．これら 86 文において，未知語処理前の文（対訳単語辞書なし）と未知語処理後の文（Model1 対訳単語辞書を使用）を，人手による対比較評価を行った．評価基準を以下に示す．

- 未知語処理 ： 未知語が適切な対訳単語に翻訳され，未知語処理後の文の意味が読み取れる
- 未知語処理 × ： 未知語が不適切な対訳単語に翻訳され，未知語処理前の文の方が意味が読み取れる
- 差なし ： 2 つの出力文の意味が読み取れない

### 6.4.4.2 人手評価結果

人手評価結果を表 6.4.3 に示す．

表 6.4.3: 翻訳実験の人手評価結果

未知語処理	未知語処理 ×	差なし
17	26	43

表 6.4.3 より，未知語処理 の数より未知語処理 × の方が多い．この結果から未知語処は困難であることが分かる．

#### 6.4.4.3 人手評価例：未知語処理 の例

翻訳実験において、未知語処理 の例を表 6.4.4～6.4.6 に示す。なお、下線は未知語処理を行う箇所を指している。

表 6.4.4: 未知語処理 の例 1

入力文	その商品 は <u>口コミ</u> 宣伝 で 売れた。
参照文	The product sold through word-of-mouth advertising .
未知語処理前	The goods were sold by <u>口コミ</u> propaganda .
未知語処理後	The goods were sold by <u>Word-of-mouth</u> propaganda .

表 6.4.5: 未知語処理 の例 2

入力文	電話 に 盗聴 器 が 取りつけて あった。
参照文	The phone had been bugged .
未知語処理前	The telephone was equipped with <u>盗聴</u> .
未知語処理後	The telephone was equipped with <u>Wiretapping</u> .

表 6.4.6: 未知語処理 の例 3

入力文	その壁 は ばい煙 で 黒ずん でいた。
参照文	The wall was dark with soot and smoke .
未知語処理前	The walls were dark with <u>ばい煙</u> .
未知語処理後	The walls were dark with <u>soot</u> .

表 6.4.4～表 6.4.6 において、未知語処理後では未知語が正しい訳語に翻訳された。未知語処理後の文の意味が読みとれるので未知語処理 とした。

#### 6.4.4.4 人手評価例：未知語処理 × の例

未知語処理 × の例を表 6.4.7～表 6.4.9 に示す。

表 6.4.7: 未知語処理 × の例 1

入力文	両法律とも最高裁判所において無効とされた。
参照文	Both laws were struck down by the Supreme Court .
未知語処理前	Both laws were <u>無効</u> in the Supreme Court .
未知語処理後	Both laws were <u>operative</u> in the Supreme Court .

表 6.4.8: 未知語処理 × の例 2

入力文	大臣の祝辞は彼が代読した。
参照文	He read the Minister's congratulatory address .
未知語処理前	The congratulatory address of the minister was <u>代読</u> him .
未知語処理後	The congratulatory address of the minister was <u>Chief</u> him .

表 6.4.9: 未知語処理 × の例 3

入力文	次の動詞の変化を示せ。
参照文	Conjugate the following verbs .
未知語処理前	<u>示せ</u> the change in the next verb .
未知語処理後	<u>Unless</u> the change in the next verb .

表 6.4.7～表 6.4.9 において，未知語処理後では未知語が誤った訳語に翻訳された．未知語が正しく翻訳できれば，意味が読みとれる文であったため未知語処理 × とした．

#### 6.4.4.5 人手評価例：差なしの例

差なしの例を表 6.4.10～表 6.4.12 示す

表 6.4.10: 差なしの例 1

入力文	彼女は激しくすすり泣いた。
参照文	She sobbed bitterly .
未知語処理前	She danced with <u>すすり泣い</u> .
未知語処理後	She danced with <u>schoolroom</u> .

表 6.4.11: 差なしの例 2

入力文	ダーツが標的の中心にいちばん近く刺さった人が勝ちだ。
参照文	The person whose dart approaches nearest the bull's-eye wins .
未知語処理前	The tip of <u>ダーツ</u> is the first one who won the game .
未知語処理後	The tip of <u>darts</u> is the first one who won the game .

表 6.4.12: 差なしの例 3

入力文	先生は青山から中野へお越しになりました。
参照文	My teacher came from Aoyama to Nakano .
未知語処理前	The teacher came from Aoyama to flower <u>青山</u> .
未知語処理後	The teacher came from Aoyama to flower <u>Aoyama-san</u> .

表 6.4.10～表 6.4.12 において、未知語処理にかかわらず文の意味が読み取れなかった。これらの文は、全て差なしと評価した。

## 第7章 おわりに

機械翻訳の問題点の1つとして、入力文中の単語が未知語として出現する問題点がある。この対策として川原らは、IBM Model1 を用いた未知語処理を提案した。この手法は、対訳学習文と IBM Model1 をもとに作成した対訳単語辞書を用いて、未知語処理を試みた。しかし、未知語処理にはまだ課題がある。この未知語処理の課題の一つに対訳単語辞書の精度が低いことが考えられる。

未知語処理に用いている対訳単語辞書に関しては、中村らに対訳単語辞書の日本語単語と英語単語の適切な対応の数を調査した。しかし、この研究では GIZA++ の対訳単語確率の計算において IBM 翻訳モデルの Model1 のみの調査で、他のモデルでの調査が行われていない。GIZA++ の対訳単語確率を計算するモデルとしては、IBM Model1 ~ Model5 や HMM などがある。また未知語の多くは、頻度 1 単語である。

本研究では、IBM Model1 ~ Model5 と HMM において頻度 1 単語の精度を調査した。具体的には、対訳学習文と GIZA++ の対訳単語確率 (IBM Model1 ~ Model5 と HMM) を用いて対訳単語を作成した。作成した対訳単語から枝刈りを行い、対訳単語辞書を作成した。そして対訳単語辞書に含まれる頻度 1 単語の精度を調査する。結果として、頻度 1 単語の精度はどのモデルも大きな差はなかった。この結果から、一番単純なモデルである IBM Model1 を使用すれば十分であると考えられる。

# 謝辞

最後に，一年間に渡り，本研究のご指導をいただきました鳥取大学工学部知能情報工学科自然言語処理研究室の村上仁一准教授，村田真樹教授に深く感謝すると共に，厚く御礼申し上げます．

また，参考にさせていただいた論文の著者の方々に対して，深く感謝申し上げます．



## 参考文献

- [1] 川原宰, 村上仁一 “日英翻訳における IBM Model1 を用いた未知語処理”, 言語学処理学会第 24 回年次大会, 2018.
- [2] 中村友哉, 村上仁一 “対訳単語辞書の精度調査”, 言語処理学会第 23 回年次大会, 2017.
- [3] Peter F.Brown, Stephen A.Della Pietra, Vincent J.Della Pietra, Robert L.Mercer: “The mathematics of statistical machine translation: Parameter Estimation”, Computational Linguistics, 1993.
- [4] Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation, “In Proceedings of the 16th conference on Computational linguistics-Volume 2”, pp. 836-841. Association for Computational Linguistics, 1996.
- [5] Frederik Jelinek. 1976. Speech Recognition by Statistical Methods. Proceedings of the *IEEE*, Vol. 64, 532-556, April 1976.
- [6] GIZA++  
<http://www.fjoch.com/GIZA++>
- [7] 川原宰, “統計翻訳における未知語処理”, 卒業論文. pp. 4-9. 2016.
- [8] 村上仁一, 藤波進 “日本語と英語の対訳文対の収集と著作権の考察”, 第一回コーパス日本語学ワークショップ, pp. 119-130. 2012.
- [9] Mecab : mecab-0.97.tar.gz , mecab-ipadic-2.7.0-20070801.tar.gz  
<http://mecab.sourceforge.net/>.
- [10] Philipp Koehn, Marcello Federico, Brooke Cowan, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, Evan Herbst, “Moses: Open Source Toolkit for Statistical Machine Translation”, Proceedings of the ACL 2007 Demo and Poster Sessions, pp. 177-180, June 2007.

- [11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate. *In Proceedings of ICLR*, 2015.
- [12] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. “Opennmt: Open-source toolkit for neural machine translation”, 2017.
- [13] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective approaches to attention-based neural machine translation”, In Proceedings of EMNLP, pp. 1412-1421, 2015.
- [14] 今仁優希, “ニューラル機械翻訳における低頻度語に処理”, 卒業論文. p.18 2018.
- [15] BLEU : Papineni Kishore, Salim Roukos, Todd Ward, Wei-Jing Zhu: “BLEU: a method for automatic evaluation of machine translation”, 40th Annual meeting of the Association for Computational Linguistics pp. 311-318, 2002.
- [16] Meter : Banerjee Satanjeev , Lavie Alon: “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”, Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-2005), pp. 65-72, June 2005.
- [17] RIBES : Hideki Isozaki, “Automatic Evaluation of Translation Quality for Distant Language Pairs”, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp.944-952. 2010.
- [18] TER : Richard Schwartz, Linnea Micciulla, John Makhoul: “A Study of Translation Edit Rate with Targeted Human Annotation”, AMTA, 2006.