概要

機械翻訳の手法として，パターン翻訳，機械翻訳等が研究されてきた．しかし，人手の翻訳には及ばない．この問題を解決するために安場らは，新たな手法として，“相対的意味論に基づく変換主導型統計機械翻訳 (TDSMT : Transfer Driven Pattern Machine Translation)”[1] を提案した．TDSMT では，学習文対と変換テーブルを用いる．まず，対訳単語を対訳単語確率 (IBM Model 1) を用いて作成する．次に，学習文対から単語レベル文パターンを作成する．さらに，新たな学習文対を用いて対訳句を選択する．最後に，順位を用いて枝刈りを行う．対訳句の順位付けは対訳単語確率によって行い，順位が低い対訳句を削除する．この手法を本実験の従来手法とする．

しかし，従来手法は対訳単語確率は高いが誤っている対訳句が存在する．その結果，誤った変換テーブルが自動作成され，誤翻訳となる．そこで本研究では，新たな変数確率 $P_{global}(X_k)$ を求めて対訳句の順位を決定し，順位が低い対訳句を削除する手法を提案する．この手法は，他の対訳句の対訳単語確率を利用する．提案手法を使用することで，変換テーブルにおける対訳句の精度向上を試みる．

実験として，従来手法と提案手法の対訳句の精度調査を行った．実験の結果，従来手法と比較して提案手法は近い精度を示した．

目 次

図 目 次

表 目 次

第1章　はじめに

機械翻訳において“相対的意味論に基づく変換主導型統計機械翻訳 (以下，TDSMT)”が提案されている [1]．TDSMT は，学習文対と変換テーブルを用いて，原言語文を入力とし，目的言語文を出力する手法である．変換テーブルは“A が B ならば C は D” で表現する． 日英翻訳の例では，A は学習文対の日本語句，B は学習文対の英語句，C は入力文の日本語句，D は出力文の英語句に当たる．対訳単語は対訳単語確率 (IBM Model 1) を用いて作成する．次に，学習文対から単語レベル文パターンを作成する．さらに，新たな学習文対を用いて対訳句を選択する．最後に，順位を用いて枝刈りを行う．対訳句の順位付けは対訳単語確率によって行い，順位が低い対訳句を削除する．

しかしこの手法では，対訳単語確率は高いが誤っている対訳句が存在する．その結果，誤った変換テーブルが自動作成され，誤翻訳となる．

そこで本稿では，新たな変数確率 $P_{global}(X_k)$ を求めて対訳句の順位を決定し，順位が低い対訳句を削除する手法を提案する．この手法は，他の対訳句の対訳単語確率を利用する．提案手法によって，従来手法と比較して変換テーブルにおける対訳句の精度向上を目指す．

実験として，従来手法と提案手法の対訳句 100 個について精度調査を行った．実験の結果，提案手法の対訳句の精度は従来手法の対訳句の精度と比較して同等だった．

本論文の構成は以下の通りである．第 2 章で従来の研究について説明し，第 3 章で提案手法について説明する．第 4 章で実験データ，実験結果と評価を示す．第 5 章で本研究の考察を述べる．

第2章　従来の研究

2.1　統計翻訳

本節は西尾ら [2] の抜粋である．

2.1.1　概要

統計翻訳とは，機械翻訳手法の一種である．原言語と目的言語の対訳文を大量に収集した対訳文より，自動的に翻訳規則を獲得し翻訳を行う．

統計翻訳には単語に基づく統計翻訳と句に基づく統計翻訳があり，初期の統計翻訳では単語に基づく統計翻訳が用いられていたが，翻訳精度は高くなかった．しかし近年，句に基づく統計翻訳が提案され，単語に基づく統計翻訳に比べて翻訳精度が高いことがわかった．このため現在は句に基づく統計翻訳が主流となっている．

2.1.2　単語に基づく統計翻訳

単語に基づく統計翻訳は単語対応の翻訳モデルを用いている．例として，ある日本語文を英語文に翻訳する場合を考える．日本語単語を英語に翻訳し，日本語単語の語順と同じ並びで英単語を並べて翻訳する．単語に基づく統計翻訳は単語対応の確率を得る IBM 翻訳モデルが用いられている．

2.1.3　IBM 翻訳モデル

IBM 翻訳モデルを以下に示す．これは，力久ら [5] の抜粋である．統計翻訳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がある．IMB 翻訳モデルは，単語に基づく統計翻訳を想定して作成された，単語対応の確率モデルである．この翻訳モデルは順に複雑な計算を行うモデル 1 から 5 の 5 つのモデルで構成される．

本章では．原言語であるフランス語文を F，目的言語である英語文を E として定義する．

IBM モデルでは，フランス語文 E，英語文 F の翻訳モデル $P(F|E)$ を計算するために，アライメント a を用いる．以下に IBM モデルの基本式を示す．

$$P(F|E) = \sum_{a} P(F, a|E) \tag{2.1}$$

アライメントとは仏単語と英単語の対応を意味している．IBM モデルのアライメントでは，各仏単語 f に対応する英単語 e は 1 つあり，各英単語 e に対応する仏単語は 0 から n 個ある．また仏単語 f において適切な英単語と対応しない場合，英語文の先頭に空単語 e_0 があると仮定し，その仏単語 f と空単語 e_0 を対応づける．

・モデル 1

(2.1) 式は以下の式に分解することができる．m はフランス語文の長さ，a_1^{j-1} はフランス語文における，1 番目から $j-1$ 番目までのアライメント，f_1^{j-1} はフランス語文における，1 番目から $j-1$ 番目まで単語を表している．

$$P(F, a|E) = P(m|E) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, f_1^{j-1}, m, E) P(f_j|a_1^{j}, f_1^{j-1}, m, E) \tag{2.2}$$

(2.2) 式ではとても複雑であるので計算が困難である．そこで，モデル 1 では以下の仮定により，パラメータの簡略化を行う．

- フランス語文の長さの確率 ϵ は m, E に依存しない

$$P(m|E) = \epsilon$$

- アライメントの確率は英語文の長さ l に依存する

$$P(a_j|a_1^{j-1}, f_1^{j-1}, m, E) = (l+1)^{-1}$$

- フランス語の翻訳確率 $t(f_j|e_{a_j})$ は，仏単語 f_j に対応する英単語 e_{a_j} に依存する

$$P(f_j|a_1^{j}, f_1^{j-1}, m, e) = t(f_j|e_{a_j})$$

パラメータの簡略化を行うことで，$P(F, a|E)$ と $P(F, E)$ は以下の式で表される．

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

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

$$= \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_{a_j}) \tag{2.5}$$

モデル 1 では翻訳確率 $t(f|e)$ の初期値が 0 以外の場合，Expectation-Maximization(EM) アルゴリズムを繰り返し行うことで得られる期待値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 翻訳確率 $t(f|e)$ の初期値を設定する．

手順 2 仏英対訳対 $(F^{(s)}, E^{(s)})$(但し，$1 \leq s \leq S$) において，仏単語 f と英単語 e が対応する回数の期待値を以下の式により計算する．

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

$\delta(f, f_j)$ はフランス語文 F 中で仏単語 f が出現する回数，$\delta(e, e_i)$ は英語文 E 中で英単語 e が出現する回数を表している．

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

1. 定数 λ_e を以下の式により計算する．

$$\lambda_e = \sum_{f} \sum_{s=1}^{S} c(f|e; F^{(s)}, E^{(s)}) \tag{2.7}$$

2. (2.7) 式より求めた λ_e を用いて，翻訳確率 $t(f|e)$ を再計算する．

$$t(f|e) = \lambda_e^{-1} \sum_{s=1}^{S} c(f|e; F^{(s)}, E^{(s)}) = \frac{\sum_{s=1}^{S} c(f|e; F^{(s)}, E^{(s)})}{\sum_{f} \sum_{s=1}^{S} c(f|e; F^{(s)}, E^{(s)})} \tag{2.8}$$

手順 4 翻訳確率 $t(f|e)$ が収束するまで手順 2 と手順 3 を繰り返す．

・モデル 2

モデル 1 では，全ての単語の対応に対して，英語文の長さ l にのみ依存し，単語対応の確率を一定としている．そこで，モデル 2 では，j 番目の仏単語 f_j と対応する英単語の位置 a_j は英語文の長さ l に加えて，j と，フランス語文の長さ m に依存し，以下のような関係とする．

$$a(a_j|j,m,l) \equiv P(a_j|a_1^{j-1}, f_1^{j-1}, m, l) \tag{2.9}$$

この関係からモデル 1 における (2.4) 式は，以下の式に変換できる．

$$\begin{aligned} P(F|E) &= \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) a(a_j|j,m,l) && (2.10) \\ &= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_{a_j}) a(a_j|j,m,l) && (2.11) \end{aligned}$$

モデル 2 では，期待値は $c(f|e;F,e)$ と $c(i|j,m,l;F,E)$ の 2 つが存在する．以下の式から求められる．

$$\begin{aligned} c(f|e;F,E) &= \frac{t(f|e)}{t(f|e_0)+\cdots+t(f|e_l)} \sum_{j=1}^{m} \delta(f,f_j) \sum_{i=1}^{l} \delta(e,e_i) && (2.12) \\ &= \sum_{j=1}^{m} \sum_{i=0}^{l} \frac{t(f|e)a(i|j,m,l)\delta(f,f_j)\delta(e,e_i)}{t(f|e_0)a(0|j,m,l)+\cdots+t(f|e_l)a(l|j,m,l)} && (2.13) \\ c(i|j,m,l;F,E) &= \sum_{a} P(a|E,F)\delta(i,a_j) && (2.14) \\ &= \frac{t(f_j|e_i)a(i|j,m,l)}{t(f_j|e_0)a(0|j,m,l)+\cdots+t(f_j|e_l)a(l|j,m,l)} && (2.15) \end{aligned}$$

$c(f|e;F,E)$ は対訳文中の英単語 e と仏単語 f が対応付けされる回数の期待値，$c(i|j,m,l;F,E)$ は英単語の位置 i が仏単語の位置 j に対応付けされる回数の期待値を表している．

モデル 2 では，EM アルゴリズムで計算すると複数の極大値が算出され，最適解が得られない可能性がある．モデル 1 では $a(i|j,m,l) = (l+1)^{-1}$ となるモデル 2 の特殊な場合であると考えられる．したがって，モデル 1 を用いることで最適解を得ることができる．

・モデル 3

モデル 3 は，モデル 1 とモデル 2 とは異なり，1 つの単語が複数対応する単語の繁殖数や単語の翻訳位置の歪みについて考慮する．またモデル 3 では単語の位置を絶対位置と

して考える．モデル 3 では以下のパラメータを用いる．

- 翻訳確率 $P(f|e)$
  英単語 e が仏単語 f に翻訳される確率
- 繁殖確率 $n(\phi|e)$
  英単語 e が ϕ 個の仏単語と対応する確率
- 歪み確率 $d(j|i,m,l)$
  英語文の長さ l，フランス語文の長さ m のとき，i 番目の英単語 e_i が j 番目の仏単語 f_j に翻訳される確率

さらに，英単語が仏単語に翻訳されない個数を ϕ_0 とし，その確率 p_0 を以下の式で求める．このとき，歪み確率は $\frac{1}{\phi_0!}$ で，$p_0 + p_1 = 1$ で p_0，p_1 は 0 より大きいとする．

$$P(\phi_0|\phi_1^l, E) = \left(\begin{array}{c} \phi_1 + \cdots + \phi_l \\ \phi_0 \end{array} \right) p_0^{\phi_1+\cdots+\phi_l-\phi_0} p_1^{\phi_0} \tag{2.16}$$

したがって，モデル 3 は以下の式で求められる．

$$\begin{aligned} P(F|E) &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} P(F,a|E) && (2.17) \\ &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \left(\begin{array}{c} m-\phi_0 \\ \phi_0 \end{array} \right) p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i! n(\phi_i|e_i) \\ &\times \prod_{j=1}^{m} t(f_j|e_{a_j}) d(j|a_j,m,l) && (2.18) \end{aligned}$$

モデル 3 では，全てのアライメントを計算するため，計算量が膨大となるので期待値を近似により求める．

・モデル 4

モデル 4 では，モデル 3 と異なり，単語の位置を絶対位置ではなく，相対位置で考える．またモデル 3 では考慮されていない各単語の位置，例えば形容詞と名詞の関係を考慮する．モデル 4 では歪み確率 $d(j|i.m,l)$ を 2 つの場合で考える．

- 繁殖数が 1 以上である英単語に対応する仏単語の中で，最も文頭に近い場合

$$P(\Pi_{[i]1} = j|\pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_1(j - \odot_{i-1}|\mathcal{A}(e_{[i-1]}), \mathcal{B}(f_j)) \tag{2.19}$$

$\odot_{i-1}$ は i-1 番目の英単語が対応する仏単語の位置を表している．

- それ以外の場合

$$P(\Pi_{[i]k} = j|\pi_{[i]1}{}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_{>1}(j - \pi_{[i]k-1}|\mathcal{B}(f_j)) \tag{2.20}$$

$\pi_{[i]k-1}$ は同じ英単語に対応している直前の仏単語を表している．

・モデル 5

モデル 4 では，単語の位置に関して直前の単語以外は考慮されていない．したがって，複数の単語が同じ位置に生じたり，単語の存在しない位置が生成される．モデル 5 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改善が施されている．

- 繁殖数が 1 以上である英単語に対応する仏単語の中で，最も文頭に近い場合

$$\begin{aligned} P(\Pi_{[i]1} &= j|\pi_1^{[i]-1}, \tau_0^l, \phi_o^l, E) \\ &= d_1(v_j|\mathcal{B}(f_j), v_{\odot_{i-1}}, v_m - \phi_{[i]} + 1)(1 - \delta(v_j, v_{j-1})) \end{aligned}$$

v_j は j 番目までの空白数，$\mathcal{A}$ は英語の単語クラス $\mathcal{B}$ はフランス語の単語クラスを表している．

- それ以外の場合

$$\begin{aligned} P(\Pi_{[i]k} &= j|\pi_{[i]1}{}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) \\ &= d_{>1}(v_j - v_{\pi_{[i]k-1}}|\mathcal{B}(f_j), v_m - v_{\pi_{[i]k-1}} - \phi_{[i]} + k)(1 - \delta(v_j, v_{j-1})) \end{aligned}$$

2.1.4 単語に基づく統計翻訳の問題点

以下に，IBM 翻訳モデルを用いて得た英日方向における単語対応の例と，日英方向における単語対応の例を示す．また，●は単語が対応した箇所を示す．

表 2.1: 英日方向の単語対応

	He	went	to	kyoto	on	business
彼	●					
は						
仕事						●
で					●	
京都				●		
に						
行っ		●				
た						

表 2.2: 日英方向の単語対応

	He	went	to	kyoto	on	business
彼	●					
は			●			
仕事						●
で					●	
京都				●		
に		●			●	
行っ		●				
た		●				

表 2.1 は日本語単語 “は” と “に” と “た” に対応する英単語が存在しない．一方で，表 2.2 は全ての単語に対して対応がとれている．単語に基づく統計翻訳は対応する単語が存在しない場合，何も無い状態から単語の発生確率を計算する．このため単語翻訳確率の信頼性が問題となっている．よって現在句に基づく統計翻訳が行われている．

2.1.5 GIZA++

GIZA++ とは，統計翻訳で用いることを前提に作られたツールである．IBM 翻訳モデルを用いて，対訳文 (原言語文と目的言語文の対) から対訳単語と単語翻訳確率を自動的に得る．

2.2 句に基づく統計翻訳

句に基づく統計翻訳は句対応の翻訳モデルを用いる. 原言語文を目的言語文に翻訳する場合に, 隣接する複数の単語 (フレーズ) を用いて翻訳を行う方法である. 本研究では日英方向の翻訳を行うため, 日英統計翻訳を説明する. 日英統計翻訳システムの枠組みを図 2.1 に示す.

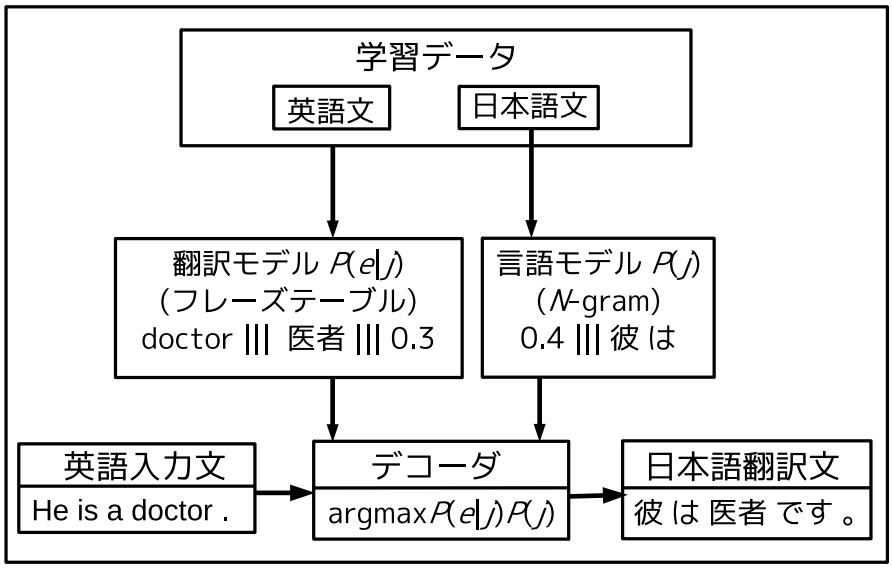

図 2.1: 日英統計翻訳の枠組み

$$E = argmax_j P(e|j) \tag{2.21}$$

$$\simeq argmax_j P(j|e)P(e) \tag{2.22}$$

ここで $P(j|e)$ は翻訳モデル，$P(e)$ は言語モデルを示す．$P(e)$ が単語であれば“単語に基づく統計翻訳” のモデル，$P(e)$ が句であれば，“句に基づく統計翻訳” のモデルとなる．

また，学習データとは対訳文 (英語文と日本語文の対) を大量に用意したものである. 学習データに含まれる各々のデータから, 翻訳モデルと言語モデルを学習する.

2.2.1 翻訳モデル

翻訳モデルとは，膨大な量の対訳データを用いて英語のフレーズが日本語のフレーズへ確率的に翻訳を行うためのモデルである．この翻訳モデルはフレーズテーブルで管理されている．以下にフレーズテーブルの例を示す.

フレーズテーブルの例

The flower ||| その 花 ||| 0.428571 0.0889909 0.428571 0.0907911 2.718
Tonight’s concert is ||| 今晩 の コンサート は ||| 0.5 0.000223681 0.5 0.0124601 2.718

左から英語フレーズ，日本語フレーズ，フレーズの英日方向の翻訳確率 $P(j|e)$，英日方向の単語の翻訳確率の積，フレーズの日英方向の翻訳確率 $P(e|j)$，日英方向の単語の翻訳確率の積，フレーズペナルティ(値は常に自然対数の底 e=2.718) である．

2.2.2　フレーズテーブル作成法

まず，GIZA++を用いて学習文から英日，日英方向の双方向で最尤な単語アライメントを得る．英日方向の単語対応の例を表 2.3，日英方向の単語対応の例を表 2.4 に示す．また，●は単語が対応した箇所を示す．

表 2.3: 日英方向の単語対応

	He	went	to	kyoto	on	business
彼	●					
は			●			
仕事						●
で					●	
京都				●		
に		●			●	
行っ		●				
た		●				

表 2.4: 英日方向の単語対応

	He	went	to	kyoto	on	business
彼	●					
は						
仕事						●
で					●	
京都				●		
に						
行っ		●				
た						

次に，得られた双方向の単語アライメントを用いて，複数単語のアライメントを得る．このアライメントは双方向の単語対応の和集合と積集合から求める．ヒューリスティックスとして双方向ともに対応する単語対応を用いる “intersection”，双方向のどちらか一方でも対応する単語対応を全て用いる “union” がある．表 2.3 と表 2.4 を用いた “intersection” の例を表 2.5，に “union” の例を表 2.6 に示す．

表 2.5: intersection の例

	He	went	to	kyoto	on	business
彼	●					
は						
仕事						●
で					●	
京都				●		
に						
行っ		●				
た						

表 2.6: union の例

	He	went	to	kyoto	on	business
彼	●					
は			●			
仕事						●
で					●	
京都				●		
に		●			●	
行っ		●				
た		●				

また “intersection” と “union” の中間のヒューリスティックスとして “grow” と “grow-diag” がある．これら 2 つのヒューリスティックスでは “intersection” の単語対応と “union” の単語対応を用いる．“grow” は縦横方向，“grow-diag” は縦横対角方向に，“intersection” の単語対応から “union” の単語対応が存在する場合にその単語対応も用いる．“grow-diag” の例を表 2.7 に示す．

表 2.7: grow-diag の例

	He	went	to	kyoto	on	business
彼	●					
は						
仕事						●
で					●	
京都				●		
に		●			●	
行っ		●				
た		●				

“grow-diag” の最後に行う処理として “final” と “final-and” がある．“final” は少なくとも片方の言語の単語対応がない場合に，“union” の単語対応を追加する．また，“final-and” は，両側言語の単語対応がない場合に，“union” の候補対応点を追加する．“grow-diag-final-and” の例を表 2.8 に示す．

表 2.8: grow-diag-final-and の例

	He	went	to	kyoto	on	business
彼	●					
は			●			
仕事						●
で					●	
京都				●		
に		●			●	
行っ		●				
た		●				

得られた単語アライメントから，全ての矛盾しないフレーズ対を得る．このとき，そのフレーズ対に対して翻訳確率を計算し，フレーズ対に確率値を付与することでフレーズテーブルを作成する．

2.2.3 言語モデル

言語モデルとは，人間が用いる言葉の自然な並びを確率としてモデル化したものであり，膨大な量の単言語データを用いて単語の列や文字の列が起こる遷移確率を付与したものである．言語モデルには以下のようなものがある．

N-gram(2.23)

統計翻訳では主に N-gram を用いる．tri-gram の式を式 2.23 に示す．

$$\sum_{i=0}^{N-1} \log_2 \frac{count(E_{i-2}, E_{i-1}, E_i)}{count(E_{i-2}, E_{i-1})} \tag{2.23}$$

E_i : 英語単語　　N : 英文の単語数
C : 対訳学習文の頻度

実際の計算例を (2.24) に示す．

$$\begin{aligned} &\log_2 P(I\ have\ a\ dog.) \\ &= \log_2 \frac{count(I\ have\ a)}{count(I\ have)} \\ &+ \log_2 \frac{count(have\ a\ dog)}{count(have\ a)} \\ &+ \log_2 \frac{count(a\ dog.)}{count(a\ dog)} \\ &= \log_2 \frac{140}{1,007} + \log_2 \frac{2}{465} + \log_2 \frac{14}{31} \\ &= -11.8545 \end{aligned} \tag{2.24}$$

High order Joint Probability(2.25)

本研究では，言語モデルに Tri-gram の代わりに High order Joint Probability を使用する. High order Joint Probability を式 2.25 に示す.

$$\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} count(J_{j-2}, J_{j-1}, J_j, E_{i-2}, E_{i-1}, E_i) \times \log_2 \frac{count(J_{j-2}, J_{j-1}, J_j, E_{i-2}, E_{i-1}, E_i)}{count(J_{j-2}, J_{j-1}, J_j) count(E_{i-2}, E_{i-1}, E_i)} \tag{2.25}$$

J_j : 日本語単語　M : 日本語文の単語数
E_i : 英語単語　　N : 英文の単語数
P : 出現確率

実際の計算例を (2.26) に示す．また，計算式が長くに及ぶため，第 1 項のみ計算例を示す.

$$\begin{aligned} &P(\text{ぶらんこが揺れている。　}\mathit{The\ swing\ is\ swinging.}) \\ &= count(\text{ぶらんこ が　}\mathit{The\ swing}) log_2 \frac{count(\text{ぶらんこ が　}\mathit{The\ swing})}{count(\text{ぶらんこ が}) P(\mathit{The\ swing})} + ... \\ &= \frac{1}{100,000} log_2 \frac{\frac{1}{100,000}}{\frac{2}{100,000}\frac{1}{100,000}} + ... \end{aligned} \tag{2.26}$$

High order Dice(2.27)

$$\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\log_2\frac{2\cdot count(J_{j-2},J_{j-1},J_j,E_{i-2},E_{i-1},E_i)}{count(J_{j-2},J_{j-1},J_j)+count(E_{i-2},E_{i-1},E_i)} \tag{2.27}$$

実際の計算例を (2.28) に示す．また，計算式が長くに及ぶため，第 1 項のみ計算例を示す．

$$\begin{aligned} &P(\text{ぶらんこが揺れている。}\quad The\ swing\ is\ swinging.)\\ &=\log_2\frac{2\cdot count(\text{ぶらんこ が}\quad The\ swing}{count(\text{ぶらんこ が})+count(The\ swing)}+...=\frac{2\cdot\frac{1}{100{,}000}}{\frac{2}{100{,}000}+\frac{1}{100{,}000}}+... \end{aligned} \tag{2.28}$$

High order Log Linear(2.29)

$$\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\log_2\left\{\frac{count(J_{j-2},J_{j-1},J_j,E_{i-2},E_{i-1},E_i)}{count(J_{j-2},J_{j-1},J_j)}\times\frac{count(E_{i-2},E_{i-1},E_i,J_{j-2},J_{j-1},J_j)}{count(E_{i-2},E_{i-1},E_i)}\right\} \tag{2.29}$$

実際の計算例を (2.30) に示す．また，計算式が長くに及ぶため，第 1 項のみ計算例を示す．

$$\begin{aligned} &P(\text{ぶらんこが揺れている。}\quad The\ swing\ is\ swinging.)\\ &=\log_2\left\{\frac{count(\text{ぶらんこ が}\quad The\ swing)}{count(\text{ぶらんこ が})}\times\frac{count(The\ swing\quad \text{ぶらんこ が})}{count(The\ swing)}\right\}\\ &=\log_2\left\{\frac{\frac{1}{100{,}000}}{\frac{2}{100{,}000}}\times\frac{\frac{1}{100{,}000}}{\frac{1}{100{,}000}}\right\} \end{aligned} \tag{2.30}$$

2.2.4 デコーダ

デコーダは，翻訳モデルと言語モデルを用いて，確率が最大となる翻訳候補を探索し，出力を行う変換器のことである．代表的なデコーダとして，“Moses” [8] がある．

入力文として “She is a teacher .” が与えられたときの翻訳例を図 2.2 に示す．

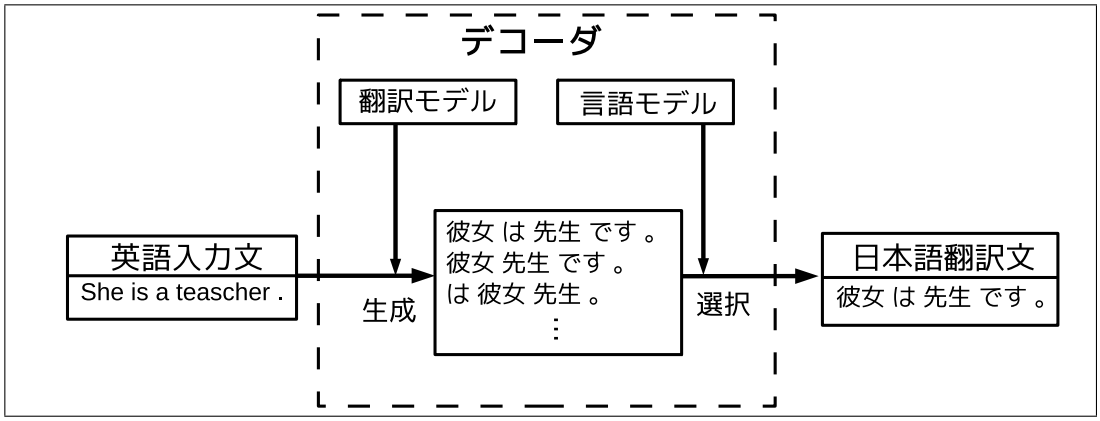

図 2.2: デコーダの動作例

日英統計翻訳において，$argmax_e P(e|j)P(j)$ の確率が最大となる英語文を出力するために，適切な順序で日本語と英語の単語対応を得る必要がある．しかし，適切な日本語文を決定するためには，計算量が膨大となり，かつ莫大な時間が必要となる．そこで計算量を削減するために，ビームサーチ法を用いる．

ビームサーチ法とは，翻訳候補の探索において，翻訳確率の低い翻訳候補を枝刈りし，探索範囲を減退する方法である．探索領域の中で一定の確率以上の翻訳候補のみを残し，それ以外の翻訳候補は除外する．

ただし，ビームサーチ法は，切り捨てられた翻訳候補が文章全体で見たときに，最大の確率を持つ翻訳候補であったという可能性がある．そのため選択した翻訳文が最適解であるとは限らないという問題がある．

2.3 相対的意味論に基づく変換主導統計機械翻訳 (TDSMT)[1]

“相対的意味論に基づく変換主導型統計機械翻訳 (TDSMT)” とは，安場らが提案した機械翻訳の手法の一種である．TDSMT は，学習文対と，変換テーブルを用いて，原言語文を入力とし，目的言語文を出力する．変換テーブルは “*A* が *B* ならば *C* は *D*” で表現する．*A* は学習文対中の原言語句，*B* は学習文対中の目的言語句，*C* は入力文中の原言語句，*D* は出力文中の目的言語句である．

原言語入力文が，学習文対の原言語側と一致するまで，入力文と変換テーブル中の *AC* を照合する．次に，一致した学習文対の目的言語側を，照合した変換テーブルの *BD* に従って変換し，目的言語翻訳文を出力する．

まず，対訳単語を対訳単語確率 (IBM Model 1) を用いて作成する．次に，学習文対から単語レベル文パターンを作成する．さらに，新たな学習文対を用いて対訳句を選択する．最後に，順位を用いて枝刈りを行う．対訳句を順位付けは対訳単語確率によって行い，順位が低い対訳句を削除する．しかし，問題点として，対訳単語確率は高いが誤っている対訳句が存在する．その結果，誤った変換テーブルが自動作成され，誤翻訳となる．

2.3.1 変換テーブル作成の手順

TDSMT では，学習文対から対訳単語と単語レベル文パターンを作成し，別の学習文対を照合することで，変換テーブルを作成する．変換テーブル作成までの手順を以下に示す．

手順 1　対訳単語の作成

学習文対と対訳単語確率 (IBM Model 1[9]) を利用して，対訳単語を作成する．このとき付与される対訳単語確率を $P_{local}(X_k)$ とする．例として，表 2.9 に示す学習文対を使用して，表 2.10 に示す対訳単語を作成する．表 2.10 の値は例であり，実際の数値とは異なる．

表 2.9: 対訳単語作成に用いる学習文対

学習文対 (日本語)	彼は彼女が好きだ。
学習文対 (英語)	He likes her.

表 2.10: 作成される対訳単語

	日本語単語	英語単語	$P_{local}(X_k)$
対訳単語 1	彼	He	0.4
対訳単語 2	好き	likes	0.5
対訳単語 3	彼女	her	0.6

手順 2　単語レベル文パターンの作成

学習文対内で対訳単語に当たる部分を変数化し，単語レベル文パターンを作成する．例を表 2.11 に示す．

表 2.11: 単語レベル文パターンの作成例

学習文対 (日本語)	彼は彼女が好きだ。
学習文対 (英語)	He likes her.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

手順 3　変換テーブルの作成

学習文対と単語レベル文パターンを照合する．変数化した対訳単語と，変数に当たる対訳句を変換テーブルとする．表 2.12 では変数 X_1 の部分から変換テーブル “「得意」が「are good at」ならば「好き」は「likes」” が得られる．

表 2.12: 変換テーブルの作成例

学習文対 (日本語)	君は料理が得意だ。
学習文対 (英語)	You are good at cooking.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .
照合する学習文対 (日本語)	彼は彼女が好きだ。
照合する学習文対 (英語)	He likes her.
変換テーブル X_1	*A*:得意　*B*:are good at *C*:好き　*D*:likes

2.3.2 $P_{local}(X_k)$ を用いた対訳句の削除

対訳句の順位付けは対訳単語確率によって行い，順位が低い対訳句を削除する．従来手法では，自分自身の対訳単語確率 ($P_{local}(X_k)$) によって選択する．$P_{local}(X_k)$ の値が高い順番で順位を決定し，順位が n 位以下であれば削除する．ここでは n=3 とし，3 位以下の対訳句を削除する．

例として表 2.13，表 2.14，表 2.15 に 3 つの学習文対と単語レベル文パターンを示す．表 2.13 は表 2.11 と同じ学習文対である．

表 2.13: 学習文対と単語レベル文パターンの例 1

学習文対 (日本語)	彼は彼女が好きだ。
学習文対 (英語)	He likes her.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

表 2.14: 学習文対と単語レベル文パターンの例 2

学習文対 (日本語)	彼はゴルフが好きだ。
学習文対 (英語)	He is keen on golf.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

表 2.15: 学習文対と単語レベル文パターンの例 3

学習文対 (日本語)	彼はギャンブルが好きだ。
学習文対 (英語)	He is so into gambling.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

また，表 2.16，表 2.17，表 2.18 に 3 つの学習文対の対訳句と対訳単語確率の例を示す．

表 2.16: 対訳句と変数確率の例 1

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$
X_0	彼	He	0.4
X_1	好き	likes	0.5
X_2	彼女	her	0.6

表 2.17: 対訳句と変数確率の例 2

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$
X_0	彼	He	0.4
X_1	好き	is keen on	0.3
X_2	ゴルフ	golf	0.7

表 2.18: 対訳句と変数確率の例 3

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$
X_0	彼	He	0.4
X_1	好き	is so into	0.4
X_2	ギャンブル	gambling	0.5

表 2.16，表 2.17，表 2.18 はいずれも単語レベル文パターンが一致している．また，*X1* について日本語句「好き」に対して異なる英語句を用いて表現されている．そこで，この 3 つの対訳句を $P_{local}(X_k)$ の値が高い順番で順位を決定し，順位が低い対訳句を削除する．それぞれの変数確率とその順位を表 2.19 に示す．

表 2.19: 各学習文対の対訳句 X_1 の変数確率とその順位 1

日本語句	英語句	$P_{local}(X_k)$	$P_{local}(X_k)$ の順位
好き	likes	0.5	1
好き	is keen on	0.3	3
好き	is so into	0.4	2

順位が 3 位以下の対訳句を削除するので，日本語句「好き」，英語句「is keen on」の対訳句を削除する．

2.3.3 問題点

対訳句を削除した後，対訳単語確率は高いが誤っている対訳句が存在する．例として 3 つの学習文対を表 2.20，表 2.21，表 2.22 に示す．表 2.20，表 2.22 はそれぞれ表 2.13，表 2.15 と同じ学習文対である．また，表 2.21 は対訳単語確率は高いが誤った対訳句が存在する学習文対である．

表 2.20: 学習文対と単語レベル文パターンの例 4

学習文対 (日本語)	彼は彼女が好きだ。
学習文対 (英語)	He likes her.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

表 2.21: 学習文対と単語レベル文パターンの例 5

学習文対 (日本語)	彼は旅行が好きだ。
学習文対 (英語)	He travels a lot.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

表 2.22: 学習文対と単語レベル文パターンの例 6

学習文対 (日本語)	彼はギャンブルが好きだ。
学習文対 (英語)	He is so into gambling.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

また，表 2.23，表 2.24，表 2.25 に 3 つの学習文対の対訳句と対訳単語確率の例を示す．

表 2.23: 対訳句と変数確率の例 4

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$
X_0	彼	He	0.4
X_1	好き	likes	0.5
X_2	彼女	her	0.6

表 2.24: 対訳句と変数確率の例 5

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$
X_0	彼	He	0.4
X_1	好き	travels	0.6
X_2	旅行	a lot	0.2

表 2.25: 対訳句と変数確率の例 6

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$
X_0	彼	He	0.4
X_1	好き	is so into	0.4
X_2	ギャンブル	gambling	0.5

表 2.23，表 2.24，表 2.25 はいずれも単語レベル文パターンが一致している．また，X_1 について日本語句「好き」に対して異なる英語句を用いて表現されている．前節と同様に，この 3 つの対訳句を $P_{local}(X_k)$ の値が高い順番で順位を決定し，順位が 3 位以下の対訳句を削除する．それぞれの変数確率とその順位を表 2.26 に示す．

表 2.26: 各学習文対の対訳句 X_1 の変数確率とその順位 2

日本語句	英語句	$P_{local}(X_k)$	$P_{local}(X_k)$ の順位
好き	likes	0.5	2
好き	travels	0.6	1
好き	is so into	0.4	3

3 位以下の対訳句を削除するので，日本語句「好き」，英語句「is so into」という対訳句を削除する．しかし，対訳句「好き」「travels」は，日本語句「好き」に対して英語句が「travels」と誤っているにも関わらず，$P_{local}(X_k)$ の値が最も高く順位が 1 となっている．このように誤った対訳句が存在する．その結果，誤った変換テーブルが自動作成され，誤翻訳となる．誤った変換テーブルの作成例を表 2.27 に示す．

表 2.27: 誤った変換テーブルの作成例

学習文対 (日本語)	君は料理が得意だ。
学習文対 (英語)	You are good at cooking.
単語レベル文パターン (日本語)	X_0 は X_2 が X_1 だ。
単語レベル文パターン (英語)	X_0 X_1 X_2 .
照合する学習文対 (日本語)	彼は旅行が好きだ。
照合する学習文対 (英語)	He travels a lot.
変換テーブル X_1	*A*:得意　*B*:are good at *C*:好き　*D*:travels

変換テーブル X_1 について，下線部の日本語句「好き」が英語句「travels」と誤っている．本稿では，この問題を解決するために新しい手法を提案する．

第3章　提案手法

3.1　概要

提案手法では，新たな変数確率 ($P_{global}(X_k)$) を求めて対訳句の順位を決定する．この手法では，他の対訳句の対訳単語確率を利用する．そして，変換テーブルにおける対訳句の精度を向上を目指す．

3.2　$P_{global}(X_k)$ の計算式

対訳句が正しい場合には，他の対訳句の対訳単語確率が高い．一方で，対訳句が誤っている場合は意訳の文であることが多いため，他の対訳句の対訳単語確率が低い．したがって，他の対訳句の対訳単語確率を利用する．

対訳句の変数 X_i の $P_{local}(X_i)$ の値を x_i，対訳変数の数を l とする．この時，対訳句 X_k の $P_{global}(X_k)$ の値を式 (1) に示す．

$$P_{global}(X_k) = \sum_{i=0}^{l-1} P_{global}(X_i)(\text{ただし、} i \neq k) \tag{3.1}$$

$P_{global}(X_k)$ の計算方法として，3 変数の場合の一般解を表 3.1 に示す．

表 3.1: 3 変数の場合の $P_{global}(X_k)$ の一般解

対訳句の変数	$P_{local}(X_k)$	$P_{global}(X_k)$
X_0	x_0	x_1+x_2
X_1	x_1	x_0+x_2
X_2	x_2	x_0+x_1

例えば，X_0 の $P_{global}(X_0)$ の値は X_1 と X_2 の $P_{local}(X_1)$ と $P_{local}(X_2)$ の値を加算する．つまり，x_1+x_2 とする．

3.3 $P_{global}(X_k)$ の計算

$P_{global}(X_k)$ の計算例を示す．表 2.20，表 2.21，表 2.22 の 3 つの学習文対について，対訳句と対訳単語確率の例を表 3.2，表 3.3，表 3.4 に示す．

表 3.2: 対訳句と変数確率の例 7

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$	$P_{global}(X_k)$
X_0	彼	He	0.4	1.1
X_1	好き	likes	0.5	1.0
X_2	彼女	her	0.6	0.9

表 3.3: 対訳句と変数確率の例 8

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$	$P_{global}(X_k)$
X_0	彼	He	0.4	0.8
X_1	好き	travels	0.6	0.6
X_2	旅行	a lot	0.2	1.0

表 3.4: 対訳句と変数確率の例 9

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$	$P_{global}(X_k)$
X_0	彼	He	0.4	0.9
X_1	好き	is so into	0.4	0.9
X_2	ギャンブル	gambling	0.5	0.8

表 3.2 で，X_0 の $P_{global}(X_0)$ は 0.5+0.6=1.1 となる．同様に，X_1 と X_2 についても計算を行う．

3.4 $P_{global}(X_k)$ を用いた対訳句の削除

表 3.2，表 3.3，表 3.4 はいずれも単語レベル文パターンが一致している．また，X_1 について日本語句「好き」に対して異なる英語句を用いて表現されている．この 3 つの対訳句を $P_{global}(X_k)$ の値が高い順番で順位を決定し，順位が n 位以下の対訳句を削除する．ここでは n=3 とする．それぞれの変数確率とその順位を表 3.5 に示す．

表 3.5: 各学習文対の対訳句 X_1 の変数確率とその順位 3

日本語句	英語句	$P_{local}(X_k)$	$P_{local}(X_k)$ の順位	$P_{global}(X_k)$	$P_{global}(X_k)$ の順位
好き	likes	0.5	2	1.0	1
好き	travels	0.6	1	0.6	3
好き	is so into	0.4	3	0.9	2

下線部の対訳句は，日本語句「好き」に対して英語句「travels」が誤っている．$P_{local}(X_k)$ を用いた順位では 1 位であったが，$P_{global}(X_k)$ を用いることで順位が 3 位となる．3 位以下を削除するので下線部の対訳句が削除される．このように $P_{global}(X_k)$ を用いることで誤った対訳句が削除される．

第4章　実験

4.1　実験目的と方法

従来手法と提案手法の対訳句を比較する．それぞれ作成された対訳句を無作為に抜き出し，精度を人手評価する．

4.2 実験条件

4.2.1 実験データ

本研究で用いる学習文対として，電子辞書などの例文より抽出した単文データを用いる [13]．データの内訳を表 4.1 に示す．

表 4.1: 実験データ

学習文対	159998 文対

学習文対の例を表 4.2 に示す．

表 4.2: 学習文対の例

学習文対	
日本語原文	英語原文
あなたに手紙がきている。	A letter is waiting for you.
昨日強振があった。	A severe earthquake was felt yesterday.
彼は株で当てた。	He struck it rich in stocks.

4.2.2 対訳句の削除に利用する順位

従来手法では，$P_{local}(X_k)$ の値が高い順番で順位を決定し，順位が n 位以下の対訳句を削除する．提案手法では，$P_{global}(X_k)$ の値が高い順番で順位を決定し，順位が n 位以下の対訳句を削除する．本実験では n=9 とする．

4.2.3 対訳句の精度調査の実験条件

従来手法と提案手法の対訳句から，それぞれ 100 個抜き出して精度を人手評価で評価する．人手評価の基準は次の 3 つである．

- ○:訳が正しい
- △:訳の正誤が判断できない
- × :訳が誤っている

4.3 実験結果

4.3.1 対訳句の削除の結果

実験から得られた対訳句の削除の例を 4.3 に示す.

表 4.3: 対訳句の削除の例

日本語句	英語句	$P_{local}(X_k)$	$P_{local}(X_k)$ の順位	$P_{global}(X_k)$	$P_{global}(X_k)$ の順位
一塁	First base	-7.2	10	-3.1	1
一塁	first	-1.1	1	-6.3	9
一塁	base	-1.8	2	-7.8	11

3 つの対訳句はいずれも日本語句「一塁」に対して異なる英語句を用いて表現されている．対訳句「一塁」「first」と対訳句「一塁」「base」は日本語と英語句が誤っているにも関わらず，$P_{local}(X_k)$ の順位が高い．一方で，対訳句「一塁」「First base」は日本語と英語句が正しいが，$P_{local}(X_k)$ の順位が低く削除される．$P_{global}(X_k)$ を用いた場合，誤っている対訳句が削除される.

4.3.2 対訳句の精度調査の結果

実験で得られた対訳句の内，それぞれ無作為に 100 個抽出し，人手評価を行った結果を表 4.4 に示す.

表 4.4: 対訳句の精度

	○	△	×
提案手法	73	9	18
従来手法	75	4	21

表 4.4 より従来手法と提案手法では対訳句の精度に差はないことが分かる．対訳句の評価例を表 4.5 に示す.

表 4.5: 対訳句の評価例

評価	日本語句	英語句
○	立場	situation
△	2 冊買っ	bought two
×	よく洗濯	launders

「立場」「situation」は訳が正しいため，評価を○とした.「2 冊買っ」「bought two」は単位が正しいか判断できないため△とした.「よく洗濯」「launders」は明らかに訳が誤っているため×とした.

第5章　考察

5.1　翻訳精度の調査

本節で，従来手法と提案手法の翻訳精度の比較を行う．翻訳精度の調査は自動評価で行う．自動評価には BLEU[10]，METEOR[11]，TER[12] を用いる．

5.1.1　実験データ

実験データの内訳を表 5.1 に示す．学習文対は表 4.2 のデータと同一である．翻訳実験に用いる入力文として，電子辞書などの例文より抽出した単文データを用いる [13].

表 5.1: 実験データの内訳

学習文対	159,998 文
入力文	100 文

入力文の例を表 5.2 に示す.

表 5.2: 入力文の例

入力文	
日本語文	参照文
ピアノの勉強にヨーロッパに行く。	Go to Europe to study the piano.
この季節としてはよい天気だ。	It's good weather for this time of year.
彼はその白馬を追いかけた。	He went after the white horse.

5.1.2 実験結果

従来手法と提案手法の実験結果を，表 5.3 に示す．

表 5.3: 翻訳精度の自動評価

	BLEU	METEOR	TER
提案手法	0.0419	0.2558	0.8548
従来手法	0.0578	0.3286	0.8256

表 5.3 より，提案手法は従来手法と比べて少し低い精度を示した．

5.2 提案手法の問題点

提案手法では，誤っているが対訳単語確率が高い対訳句を削除することができる．しかし，対訳句の削除を行う前に誤っている対訳句が存在しない場合，より正しい対訳句を削除する場合がある．例として 3 つの学習文対を表 5.4，表 5.5 に示す．

表 5.4: 学習文対と単語レベル文パターンの例 7

学習文対 (日本語)	私はペンを取る。
学習文対 (英語)	I take the pen.
単語レベル文パターン (日本語)	X_0 は X_2 を X_1。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

表 5.5: 学習文対と単語レベル文パターンの例 8

学習文対 (日本語)	彼はそれを取る。
学習文対 (英語)	He takes it.
単語レベル文パターン (日本語)	X_0 は X_2 を X_1。
単語レベル文パターン (英語)	X_0 X_1 X_2 .

また，表 5.6，表 5.7 に 2 つの学習文対の対訳句と対訳単語確率の例を示す．

表 5.6: 対訳句と変数確率の例 10

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$	$P_{global}(X_k)$
X_0	私	I	0.4	1.1
X_1	取る	take	0.6	0.9
X_2	ペン	the pen	0.5	1.0

表 5.7: 対訳句と変数確率の例 11

対訳句の変数	日本語句	英語句	$P_{local}(X_k)$	$P_{global}(X_k)$
X_0	彼	He	0.4	1.1
X_1	取る	takes	0.5	1.0
X_2	それ	it	0.6	0.9

表 5.4，表 5.5 は単語レベル文パターンが一致している．また，X_1 について日本語句「取る」に対して異なる英語句を用いて表現されている．この 2 つの対訳句を $P_{global}(X_k)$ の値が高い順番で順位を決定し，順位が n 位以下の対訳句を削除する．ここでは n=2 とする．それぞれの変数確率とその順位を表 5.8 に示す．

表 5.8: 各学習文対の対訳句 X_1 の変数確率とその順位 4

日本語句	英語句	$P_{local}(X_k)$	$P_{local}(X_k)$ の順位	$P_{global}(X_k)$	$P_{global}(X_k)$ の順位
取る	take	0.6	1	0.9	2
取る	takes	0.5	2	1.0	1

対訳句「取る」「take」は $P_{local}(X_k)$ を用いた順位では 1 位となるが，$P_{global}(X_k)$ を用いた順位では 2 位となり削除される．一方で，対訳句「取る」「takes」は $P_{local}(X_k)$ を用いた順位では 2 位であるが，$P_{global}(X_k)$ を用いた順位では 1 位となる．このように提案手法を用いることで，より正しい対訳句が削除される場合がある．

5.3 従来手法と提案手法を組み合わせた翻訳精度

前述の通り，従来手法と提案手法にそれぞれ問題点が存在する．したがって，従来手法と提案手法を組み合わせた対訳句の順位付けと削除が，翻訳精度の向上に有効であると考える．そこで，$P_{local}(X_k)$ の順位が n 位以下の対訳句，または $P_{global}(X_k)$ の順位が n 位以下の対訳句を削除する．ここでは n=9 とする．

5.3.1 実験データ

実験データの内訳を表 5.9 に示す．学習文対と入力文は表 5.1 のデータと同一である．

表 5.9: 実験データの内訳 2

学習文対	159,998 文
入力文	100 文

5.3.2 実験結果

表 5.3 と比較した実験結果を，表 5.10 に示す．

表 5.10: 翻訳精度の自動評価 2

	BLEU	METEOR	TER
提案手法	0.0419	0.2558	0.8548
従来手法	0.0578	0.3286	0.8256
従来手法+提案手法	0.0270	0.2280	0.8717

表 5.10 より，従来手法+提案手法は従来手法や提案手法と比べて低い精度を示した．

5.4 閾値を変更した場合の翻訳精度

表 4.4 と表 5.10 の結果より，精度が向上しない原因を考察する．削除前の対訳句において，誤っているが対訳単語確率が高い対訳句の数が少ないことが挙げられる．そのため，表 4.4 では，100 個の対訳句の精度に変化がない．また，表 5.10 では，正しい対訳句が多数削除されることで精度が下がる．そこで，$P_{global}(X_k)$ の閾値を変更する．そして，再度実験を行い翻訳精度の向上を目指す．

5.4.1 実験データ

実験データの内訳を表 5.11 に示す．学習文対と入力文は表 5.1 のデータと同一である．

表 5.11: 実験データの内訳 3

学習文対	159,998 文
入力文	100 文

5.4.2 対訳句の削除に利用する順位

$P_{local}(X_k)$ の閾値として，順位が n 位以下の対訳句を削除する．ここでは n=9 とする．また，$P_{global}(X_k)$ の閾値として，m 位以下の対訳句を削除する．m の値を変更し，実験を行う．

5.4.3 実験結果

実験結果を，表 5.12 に示す．

表 5.12: 翻訳精度の自動評価 3

m	BLEU	METEOR	TER
8	0.0419	0.2558	0.8548
512	0.0501	0.2940	0.8230
1024	0.0580	0.3047	0.8186
4096	0.0589	0.3272	0.8189
30000	0.0594	0.3357	0.8114
45000	0.0606	0.3355	0.8055

表 5.12 より，$P_{global}(X_k)$ の閾値を変更することで，翻訳精度の向上が確認できた．m の値を大きくすると，翻訳精度も同時に向上した．したがって，誤っているが対訳単語確率が高い対訳句の数が少ないと考えられる．

第6章　おわりに

本研究では，新たな変数確率を求めて順位付けを行い，対訳句の精度を向上する方法を提案した．実験結果より，提案手法は従来手法と同等の精度を示した．追加実験では，従来手法と提案手法を組み合わせて翻訳精度を調査した．実験結果より，翻訳精度が向上した．今後は翻訳精度を更に向上させていきたい．

謝辞

本研究を進めるにあたり，研究の説明や論文の書き方など様々なご指導を頂きました鳥取大学工学部電気情報系工学科自然言語処理研究室の村上仁一准教授に心から御礼申し上げます．また，本研究を進めるにあたり，御指導，御助言を頂きました，村田真樹教授に心から御礼申し上げます．また，自然言語処理研究室の皆様へ心から感謝の気持ちと御礼を申し上げたく，謝辞にかえさせて頂きます．

参考文献

[1] 安場裕人， 村上仁一， “変換主導型翻訳の提案”, 自然言語処理学会第 24 年次大会, March，2018.

[2] 西尾聡一郎，“パターンに基づく統計翻訳における文パターン確率の考察”，平成 27 年度 卒業論文，pp.3-16，February 2016.

[3] Franz Josef Och, Hermann Ney, “A Systematic Comparison of Various Statistical Alignment Models”, Computational Linguistics, 29(1), pp.299-314, 1996.

[4] 江木孝史, 村上仁一, 徳久雅人, “句に基づく対訳文パターンの自動作成と統計的手法を用いた英日パターン翻訳”, 自然言語処理学会第 20 回年次大会予稿集, pp.951-954, 2014.

[5] 力久 剛士，“レーベンシュタイン距離を用いた翻訳精度の向上”，平成 26 年度 卒業論文，pp.3-15，February 2015.

[6] Vladimir Iosifovich Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals”, Soviet Physics Doklady, 10(8), pp.707-710, 1966.

[7] 松本大輝，村上仁一. “翻訳における分野依存性を軽減する言語モデルの調査” 自然言語処理学会第 25 回年次大会， March 2019.

[8] 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.

[9] 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.

[10] BLEU: “a Method for Automatic Evaluation of Machine Translation” , Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics(ACL), pp.311-318. 2002.

[11] METEOR; Lavie Alon, and Denkowski Michael “An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgemnts”, Proceedings of the Second Workshop on Statistical Machine Translation, pp.228-231. 2007.

[12] Richard Schwartz, Linnea Micciulla, John Makhoul: “A Study of Translation Edit Rate with Targeted Humman Annotation”, AMTA, 2006.

[13] 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予稿集, pp.119-130, 2012.