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機械翻訳 (TDSMT : Transfer Driven Machine Translation)”[1] を提案した．この手法を本実験の従来手法とする．しかし，TDSMT はカバー率 (出力文数/入力文数) が低いという問題があった．この問題の原因の一つは，TDSMT で出力文を得るためには，変換テーブルを適用した入力文が，学習文対に完全に一致する必要しなければならないこと，にある．そこで本研究では，“相対的意味論に基づく変換主導統型パターン統計機械翻訳 (TDPBSMT : Transfer Driven Pattern Based Machine Translation)” を提案する．提案手法を使用することで，カバー率の向上を試みる．実験として，500 文を入力としたカバー率 (出力文数/入力文数) の調査を行った．その結果，提案手法がカバー率 96.6%，従来手法がカバー率 33.2% という結果になった．実験の結果，従来手法と比較して提案手法はカバー率が向上した．

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．

しかしこの手法で出力文を得るためには，変換テーブルを適用した，入力文が学習文対に完全に一致する必要がある．従って，入力文数に対して，得られる出力文数が少ないという問題がある．

そこで本稿では，“相対的意味論に基づく変換主導統型パターン統計機械翻訳 (以下，TDPBSMT)” を提案する．この手法は，変換テーブルを “*A* が *B*” と “*C* が *D*” の 2 つに分割する．次に，“*A* が *B*” を利用して文パターンを作成する．そして，“*C* が *D*” を文パターンに適用する．提案手法によって，従来手法と比較して出力文数が向上する．

実験として，500 文を入力としたカバー率 (出力文数/入力文数) の調査を行った．また，翻訳精度の調査として，100 文の対比較調査を行った．実験の結果，提案手法のカバー率は従来手法と比較して向上した．また，翻訳精度は差がなかった．

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 TDPBSMT 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.

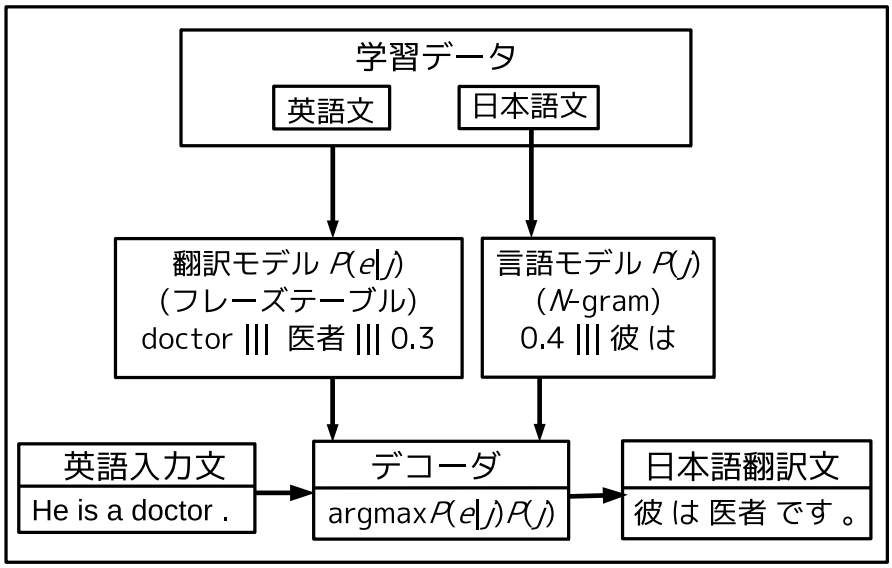

図 2.1: 日英統計翻訳の枠組み

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

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

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.

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

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．

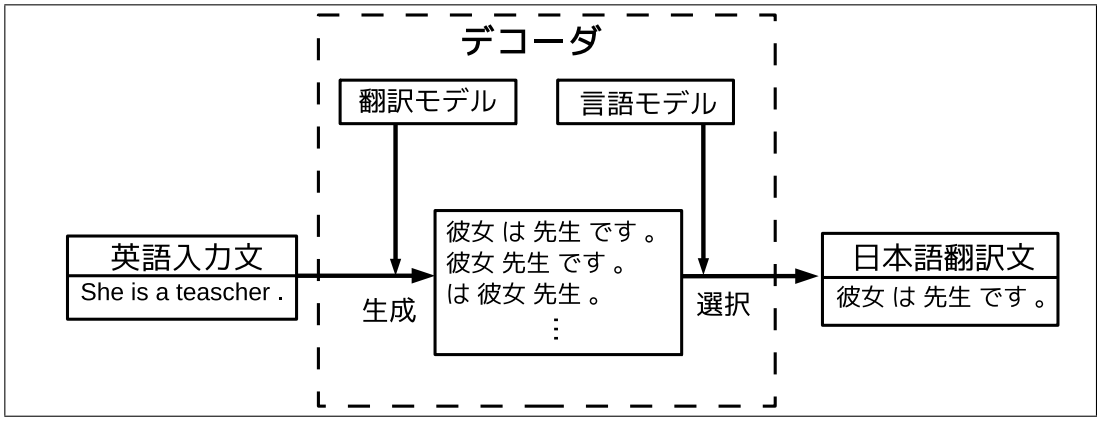

図 2.2: デコーダの動作例

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．

TDSMT は適切な学習文対及び，変換テーブルが存在した場合，翻訳精度の高い出力文を得ることができる．しかし，TDSMT は変換テーブルを適用した，入力文が学習文対に完全に一致しない場合は翻訳ができない．従って，問題点として，入力文に対するカバー率が低い．

2.3.1 TDSMT の手順

TDSMT の手順を示す．手順は “学習” と “翻訳” の二部からなる．

2.3.2 学習の手順

TDSMT における学習は “変換テーブルの作成” のみである．本節で作成手順を示す．

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_w 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 (日本語側)	彼の弟は学生だ。
学習文対 (英語側)	His brother is a student.

表 2.10: 作成される対訳単語

	日本語単語	英語単語	p_w
対訳単語 1	彼	His	0.4
対訳単語 2	弟	brother	0.7
対訳単語 3	学生	student	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 (日本語側)	彼の兄は医者だ。
学習文対 (英語側)	His brother is a doctor.
単語レベル文パターン (日本語側)	*X0* の *X1* は *X2* だ
単語レベル文パターン (英語側)	*X0 X1* is a *X2*

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 *N2* の部分から変換テーブル “「学生」が「student」ならば「教師」は「teacher」” が得られる．

表 2.12: 変換テーブルの作成例

学習文対 (日本語側)	彼の弟は学生だ。
学習文対 (英語側)	His brother is a student.
単語レベル文パターン (日本語側)	*X0* の *X1* は *X2* だ。
単語レベル文パターン (英語側)	*X0 X1* is a *X2* .
照合する学習文対 (日本語側)	私の母は教師だ。
照合する学習文対 (英語側)	My mother is a teacher.
変換テーブル *(X2)*	*A*:学生　*B*:student *C*:教師　*D*:teacher

手順 4　変換テーブルに確率を付与

対訳単語確率 P_w を利用し，変換テーブルに確率を付与する．この確率を変換テーブル確率 P_v とする．

1. 変換テーブルの *CD* に存在する全ての日英単語の組み合わせを確認する．
2. 日本語単語に対応する英語単語の中で，対訳単語確率 P_w の最大値を得る．
3. 各日本語単語について得られた値と，変換テーブルの *AB* の対訳単語確率 P_w について，対数の総和を求める．

2.3.3 翻訳の手順

本節で TDSMT における翻訳の手順を示す．入力文を「私の姉は教師だ。」とする．

手順 1　入力文に日本語側の変換テーブルを適用

変換テーブルの C と A を利用して，入力文を学習文対の日本語側と一致させる．表 2.13 では入力文中の「教師」を「生徒」に変換する．

表 2.13: 日本語側変換テーブルの適用例

入力文	私の姉は教師だ。
変換テーブル:C	教師
変換テーブル:A	生徒
一致する学習文対 (日本語側)	私の姉は生徒だ。

手順 2　学習文対に英語側の変換テーブルを適用

手順 1 と同じ変換テーブルの B と D を学習文対の英語側に適用し，出力候補文を作成する．表 2.14 では学習文対中の「student」を「teacher」に変換している．

表 2.14: 英語変換テーブルの適用例

一致した学習文対 (日本語側)	私の姉は生徒だ。
一致した学習文対 (英語側)	My sister is a student.
変換テーブル:B	student
変換テーブル:D	teacher
出力候補文	My sister is a teacher.

手順 3　最終的な出力文の決定

複数の出力候補文が得られた場合，計算式 (2.31) に従って，最終的な出力文を決定する．ここで P_m は言語モデルの確率である．

$$logP = logP_v + logP_m \tag{2.31}$$

図 2.3 に TDSMT の流れ図を示す.

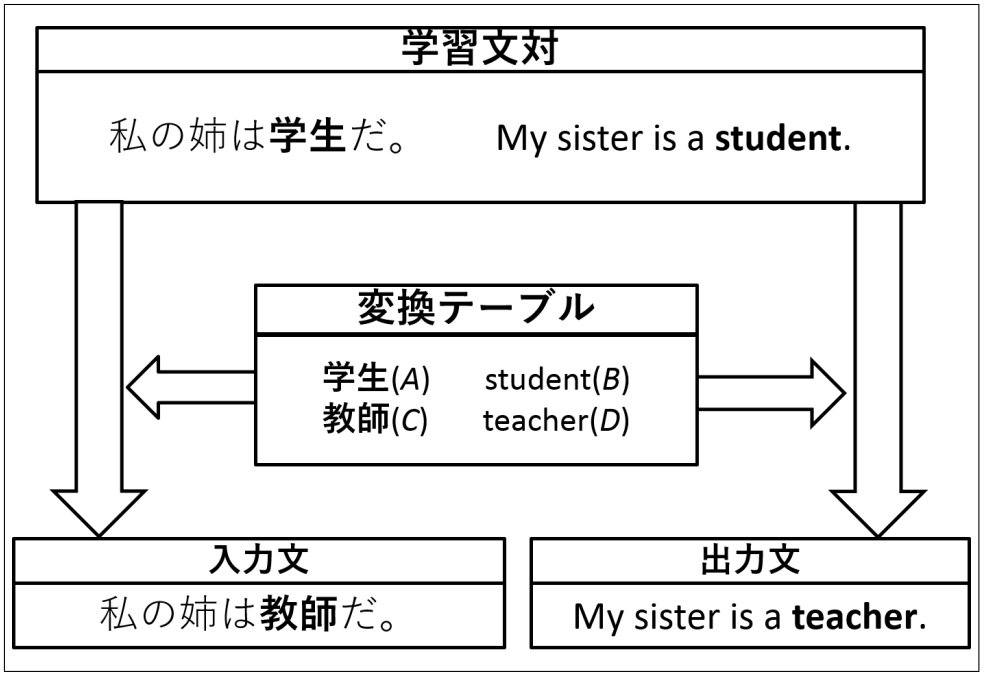

図 2.3: TDSMT の流れ図

2.3.4 問題点

従来手法の問題点は，入力文に対するカバー率が低いことである．出力不可能な例として表 2.15，2.16 があげられる．

表 2.15: 出力不可能な例

入力文	その数値は小さくなった。
一致させたい学習文対 (日本語側)	その振幅は大きくなった。
一致させたい学習文対文対 (英語側)	The amplitude increased.
想定する出力文	The value was small.

表 2.16: 用意されている変換テーブル

	A	*B*	*C*	*D*
変換テーブル 1	振幅	amplitude	数値	value
変換テーブル 2	大きくな	increased	走る	run
変換テーブル 3	大きくな	grow	小さくな	was small

上記の例が出力不可能となる流れを説明する．

1. 変換テーブル 1 を適用し入力文中の「数値」を「振幅」に変換する．
2. 変換テーブル 2 は「C : 走る」が学習文対の日本語側に存在しない．
3. 変換テーブル 3 は「B : grow」が学習文対の英語側に存在しない．

以上より，示した例は出力不可能である．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　相対的意味論に基づく変換主導型パターンベース統計機械翻訳 (TDPBSMT)

3.1　TDPBSMT の概要

“相対的意味論に基づく変換主導型パターンベース統計機械翻訳” では文パターンを用いて翻訳を行う．この手法は，学習文対の代わりに文パターンを利用するので，出力文数が増加する．

3.2　TDPBSMT の手順

本節で TDPBSMT の手順を示す．手順の説明で用いる入力文と，作成する変換テーブルと，用意されている学習文対は，表 2.15，2.16 と同一とする．

3.2.1 学習の手順

TDPBSMT における学習は「変換テーブルの作成」と「文パターンの作成」である．本節で作成手順を示す.

手順 1　変換テーブルを作成

TDSMT と同様の方法で変換テーブルを作成する.

手順 2　変換テーブルを分割

変換テーブルを変換テーブル $_{AB}$ と変換テーブル $_{CD}$ に分割する．表 3.1 では 3 つの変換テーブルを分割している.

表 3.1: 変換テーブルの分割

変換テーブル $_{AB}1$	A : 振幅	B : amplitude
変換テーブル $_{CD}1$	C : 数値	D : value
変換テーブル $_{AB}2$	A : 大きくな	B : increased
変換テーブル $_{CD}2$	C : 走る	D : run
変換テーブル $_{AB}3$	A : 大きくな	B : grow
変換テーブル $_{CD}3$	C : 小さくな	D : was small

手順 3　文パターンを作成

学習文対の，変換テーブル $_{AB}$ に当たる単語を変数化し，文パターンを作成する．表 3.2 では 2 つの対訳単語を変数に置き換えている．

表 3.2: 文パターンの作成例

	日本語側	英語側
学習文対	その振幅は大きくなった。	The amplitude increased.
変換テーブル $_{AB}1$	*A* : 振幅	*B* : amplitude
変換テーブル $_{AB}2$	*A* : 大きくな	*B* : increased
作成される文パターン	その *X0* は *X1* った。	The *X0 X1*.

手順 4　文パターンに確率を付与

対訳単語確率 P_w を利用して，文パターンに確率を付与する．この確率を文パターン確率 P_p とする．以下にその手順を示す．

1. 文パターンに存在する全ての日英単語の組み合わせを確認する．
2. 日本語単語に対応する英語単語の中で，対訳単語確率 P_w の最大値を得る．
3. 各日本語単語について得られた値について，対数の総和を求める．

3.2.2 翻訳の手順

本節で翻訳の手順を示す.

手順 1　入力文に日本語側の変換テーブルを適用

入力文中の変換テーブル$_{CD}$に存在する語句を変数に変換し，文パターンに一致させる. 表 3.3 に例を示す.

表 3.3: 入力文への変換テーブル$_{CD}$適用例

入力文	その数値は小さくなった。
変換テーブル$_{CD}$1	*C* : 数値
変換テーブル$_{CD}$3	*C* : 小さくな
一致する文パターン (日本語側)	その *X0* は *X1* った。

手順 2　文パターンに変換テーブルの英語側を適用

文パターンの変数部に変換テーブル$_{CD}$の英語側を代入し，出力候補文を作成する. 表 3.4 に例を示す.

表 3.4: 文パターンへの変換テーブル$_{CD}$適用例

一致した文パターン (日本語側)	その *X0* は *X1* った。
一致した文パターン (英語側)	The *X0 X1*.
変換テーブル$_{CD}$1	*D* : value
変換テーブル$_{CD}$3	*D* : was small
出力候補文	The value was small.

手順 3　最終的な出力文の決定

複数の出力候補文が得られた場合，計算式 (3.1) に従って最終的な出力文を決定する.

$$logP = logP_v + logP_p + logP_m \tag{3.1}$$

図 3.1 に TDPBSMT の流れ図を示す.

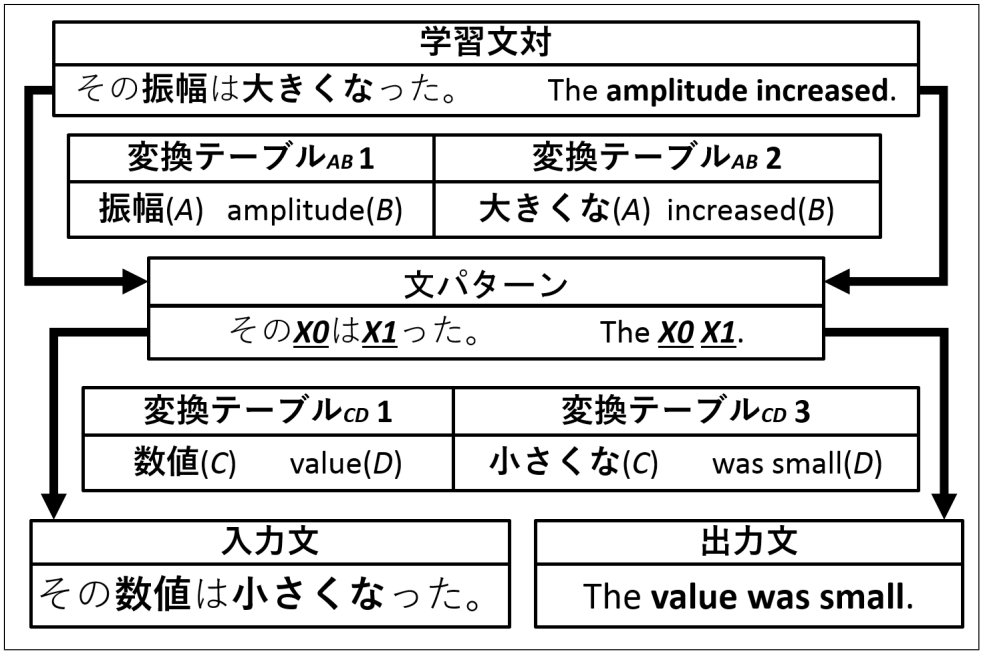

図 3.1: TDPBSMT の流れ図

また，実際の出力例を表 3.5 に示す.

表 3.5: TDPBSMT の出力例

入力文	視界は極めて良好であった。
文パターン (日本語側)	*X2* は *X0 X1* であった 。
文パターン (英語側)	The *X2* was *X0 X1* .
変換テーブル $_{CD}$(*X0*)	*C* : 極めて　*D* : extremely
変換テーブル $_{CD}$(*X1*)	*C* : 良好　*D* : good
変換テーブル $_{CD}$(*X2*)	*C* : 視界　*D* : sight
出力文	The sight was extremely good.

第4章　実験

4.1　実験目的と方法

TDPBSMT と TDSMT を比較する．入力文数に対する翻訳可能な文の割合 (カバー率) と，翻訳精度を調査する．翻訳精度の調査は自動評価と人手評価で行う．自動評価には BLEU[10]，METEOR[11]，TER[12] を用いる．

4.2 実験条件

4.2.1 実験データ

本研究では，学習文対および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]．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	160,000 文対
入力文	500 文

学習文対および，入力文の例を表 4.2 と表 4.3 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
あいつは甘えている。	He is spoiled.
あの小川で釣りをしよう。	Let's fish the creek.
いつかは石油資源が枯渇する。	Oil resources will dry up someday.

表 4.3: 入力文の例

入力文	
日本語文	参照文
花は太陽の方に傾く。	Flowers bend toward the sun.
彼らの間に不満が増大した。	Discontent waxed among them.
彼はすぐ帰国の途につく。	He will soon start for home.

TDPBSMT で作成された変換テーブルと，文パターンの総数を表 4.4 に示す．

表 4.4: 変換テーブルと文パターンの総数

変換テーブル	5,058,468 個	文パターン	165,051 個

4.2.2 カバー率の調査の実験条件

TDPBSMT と TDSMT を用いて，入力文を 500 文として，得られた出力文数を調査する．

4.2.3 翻訳精度の調査の実験条件

TDPBSMT と TDSMT を用いて，得られた出力文を自動評価と人手評価で評価する．また，人手評価は対比較評価で行う．人手評価の基準は次の 4 つである．

- TDPBSMT ○ : TDPBSMT の方が優れている．
- TDSMT ○ : TDSMT の方が優れている．
- 差なし : それぞれの翻訳結果は異なるが，
  翻訳精度には差がない．
- 同一 : それぞれの翻訳結果が一致している．

4.3 実験結果

4.3.1 カバー率調査の結果

カバー率調査の結果を 4.5 に示す．表中のかっこ内の数値は出力文数である．

表 4.5: カバー率調査結果 (500 文)

TDPBSMT	TDSMT
96.6%(483 文)	33.2%(166 文)

表 4.5 より TDPBSMT は TDSMT と比較して, カバー率が向上したことが分かる．

4.3.2 翻訳精度の調査の結果

4.3.2.1 自動評価の結果 (100 文)

TDSMT で出力を得られた文の内，100 文を対象に自動評価を行った結果を表 4.6 に示す.

表 4.6: 自動評価結果 (100 文)

	BLEU	METEOR	TER
TDPBSMT	0.23	0.48	0.57
TDSMT	0.18	0.43	0.62

表 4.6 より TDSPBSMT と TDSMT の翻訳精度にほぼ差はない.

4.3.2.2 人手評価の結果 (100 文)

TDSMT で出力を得られた文の内，100 文を対象に対比較評価を行った結果を表 4.7 に示す.

表 4.7: 人手評価結果 (100 文)

TDPBSMT ○	22
TDSMT ○	21
差なし	44
同一	13

表 4.7 より TDPBSMT と TDSMT の翻訳精度に差はないことが分かる．評価例を表 4.8，表 4.9，表 4.10，表 4.11，表 4.12，表 4.13，表 4.14，表 4.15，表 4.16 に示す．表中の下線部は評価の根拠とした部分である.

表 4.8: TDPBSMT ○とした例 1

入力文	その演説は拍手によって中断された。
TDPBSMT	The speech was interrupted by applause.
TDSMT	The speech was was interrupted according applause.
参照文	The speech was interrupted by applause.

表 4.8 の TDPBSMT と TDSMT の出力文を比較すると，TDPBSMT の出力文は参照文と同一である．TDSMT は be 動詞「was」が連続している．これは文法上不適当である．従って，TDPBSMT が TDSMT より優れていると判断した．

表 4.9: TDPBSMT ○とした例 2

入力文	戦闘が本格的に始まった。
TDPBSMT	The battle began in earnest.
TDSMT	The battle began in dead earnest.
参照文	The battle began in real earnest.

表 4.9 の TDPBSMT と TDSMT の出力文を比較すると，TDPBSMT は入力文と同様の意味を読み取れる．TDSMT は「dead」という入力文の意味と合致しない単語が含まれている．従って，TDPBSMT が TDSMT より優れていると判断した．

表 4.10: TDPBSMT ○とした例 3

入力文	鉄の神経をしている。
TDPBSMT	He has an iron nerves.
TDSMT	The iron is has nerves.
参照文	She has iron nerves.

表 4.10 の TDPBSMT と TDSMT の出力文を比較すると，TDPBSMT は入力文と同様の意味を読み取れる．TDSMT は入力文と同様の意味を読み取れない．従って，TDPBSMT が TDSMT より優れていると判断した．しかし，TDPBSMT の出力文も，「an」に続く名詞句が，複数形の「iron nerves」といった不良な点がある．

表 4.11: TDSMT ○とした例 1

入力文	彼女はベッドで寝返りを打った。
TDPBSMT	I turned over in bed.
TDSMT	She turned over in bed.
参照文	She rolled in the bed.

表 4.11 の TDPBSMT と TDSMT の出力文を比較すると，TDPBSMT は主語が入力文と異なる．TDSMT は入力文と同様の意味を読み取れる．従って，TDSMT が TDPBSMT より優れていると判断した．

表 4.12: TDSMT ○とした例 2

入力文	その翌日のバスの切符を買った。
TDPBSMT	He bought a ticket for the next bus.
TDSMT	He bought a ticket for bus from the day.
参照文	He booked himself for the following day's bus.

表 4.12 の TDPBSMT と TDSMT の出力文を比較すると，TDPBSMT は「次のバスの切符を買った。」と読める出力文となっている．TDSMT は入力文と同様の意味を読み取れる．従って，TDSMT が TDPBSMT より優れていると判断した．

表 4.13: TDSMT ○とした例 3

入力文	設備のよいホテルに泊まった。
TDPBSMT	We stayed at the hotel good equipment.
TDSMT	I stayed at a good of equipment.
参照文	I stayed a hotel that was well equipped and well furnished.

表 4.13 の TDPBSMT と TDSMT の出力文を比較すると，TDPBSMT は「ホテルに泊まった」と読み取れない．TDSMT は入力文と同様の意味を読み取れる．従って，TDSMT が TDPBSMT より優れていると判断した．

表 4.14: 差なしとした例 1

入力文	彼らはフランスに使者をやった。
TDPBSMT	They did a messenger to the French.
TDSMT	They gave aa messenger to French.
参照文	They sent a messenger France.

表 4.15: 差なしとした例 2

入力文	その対立は深まった。
TDPBSMT	The conflict has grown.
TDSMT	The comflict has deepened.
参照文	The antagonism deepened.

表 4.16: 差なしとした例 3

入力文	腕に覚えがある。
TDPBSMT	These events caused recession.
TDSMT	There recall arm.
参照文	Use a fountain pen or a ball point pen.

第5章　考察

5.1　TDPBSMT と Moses の翻訳精度の比較

本節で，提案した TDPBSMT と，一般に利用される句に基づく機械翻訳 (Moses)[8] の翻訳精度の比較を行う．

5.1.1　実験データ

実験データの内訳を表 5.1 に示す．学習文対と入力文は，表 4.2 のデータと同一である．

表 5.1: 実験データの内訳

学習文対	160,000 文
ディベロップメント文	1,000 文
入力文	100 文

5.1.2 実験結果

TDPBSMT と Moses の実験結果を，表 5.2 と表 5.3 に示す.

表 5.2: TDPBSMT と Moses の自動評価 (100 文)

	BLEU	METEOR	TER
TDPBSMT	0.20	0.49	0.60
Moses	0.24	0.49	0.56

表 5.3: TDPBSMT と Moses の人手評価結果 (100 文)

TDPBSMT ○	26
Moses ○	22
差なし	36
同一	16

表 5.2，表 5.3 より TDPBSMT と TDSMT の翻訳精度にほぼ差はないことが分かる．評価例を表 5.4，表 5.5，表 5.6，表 5.7，表 5.8，表 5.9，表 5.10，表 5.11，表 5.12 に示す．表中の下線部は評価の根拠とした部分である．

表 5.4: TDPBSMT ○とした例 1

入力文	彼は机を叩いた。
TDPBSMT	He knocked on the desk.
Moses	He was on the desk.
参照文	He pounded on the desk.

表 5.4 の TDPBSMT と Moses の出力文を比較すると，TDPBSMT は入力文と同様の意味が読み取れる．Moses は動詞が入力文と食い違う．従って，TDPBSMT が Moses より優れていると判断した．

表 5.5: TDPBSMT ○とした例 2

入力文	彼は顔をまともに打たれた。
TDPBSMT	He was struck straight in the face.
Moses	He was struck with the eye.
参照文	The blow struck him full in the face.

表 5.5 の TDPBSMT と Moses の出力文を比較すると，TDPBSMT は入力文と同様の意味が読み取れる．Moses は下線部の単語が入力文の意味と異なる．従って，TDPBSMT が Moses より優れていると判断した．

表 5.6: TDPBSMT ○とした例 3

入力文	これは規格から外れている。
TDPBSMT	This is off the standard.
Moses	This is the standard.
参照文	This is excluded from the specifications.

表 5.6 の TDPBSMT と Moses の出力文を比較すると，TDPBSMT は入力文と同様の意味が読み取れる．Moses は入力文と意味が異なる．その原因は下線部の単語の有無である．従って，TDPBSMT が Moses より優れていると判断した．

表 5.7: Moses ○とした例 1

入力文	彼女はベッドで寝返りを打った。
TDPBSMT	I turned over in bed.
Moses	She turned over in bed.
参照文	She rolled in the bed.

表 5.7 の TDPBSMT と Moses の出力文を比較すると，TDPBSMT は，名詞が入力文と異なる．Moses は入力文と同様の意味が読み取れる．従って，Moses が TDPBSMT より優れていると判断した．

表 5.8: Moses ○とした例 2

入力文	りんごが枝もたわわになっている。。
TDPBSMT	The apples the branches.
Moses	The apples were heavily laden with branch.
参照文	The apple trees are heavy with fruit.

表 5.8 の TDPBSMT と Moses の出力文を比較すると，TDPBSMT は動詞が存在せず，入力文の意味が読み取れない．Moses は入力文と同様の意味が読み取れる．従って，Moses が TDPBSMT より優れていると判断した．

表 5.9: Moses ○とした例 3

入力文	体重の問題を抱えている。
TDPBSMT	The weight of the problem.
Moses	I have a question of weight.
参照文	He has a weight problem.

表 5.9 の TDPBSMT と Moses の出力文を比較すると，TDPBSMT は動詞が存在せず，入力文の意味が読み取れない．Moses は入力文と同様の意味が読み取れる．従って，Moses が TDPBSMT より優れていると判断した．

表 5.10: 差なしとした例 1

入力文	急ブレーキをかける。
TDPBSMT	Put on the brakes suddenly.
Moses	Place the brakes sharply.
参照文	To step on the brakes suddenly.

表 5.11: 差なしとした例 2

入力文	裁判官はその異議を却下した。
TDPBSMT	The judge dismissed objection.
Moses	The judge dismissed the dissent.
参照文	The judge overruled the objection.

表 5.12: 差なしとした例 3

入力文	その対立は深まった。
TDPBSMT	The conflict has deepened.
Moses	The conflict has grown.
参照文	The antagonism deepened.

5.2　一般的な機械翻訳手法と比較した提案手法の利点

提案手法は翻訳の手順を詳しく追跡することが可能である．このため，出力結果の解析が容易である．一方，現在主流となっているニューラル機械翻訳や Moses などでは，詳細な出力結果の解析が困難である．

5.3 提案手法の問題点

本節では提案手法の問題点を考察する.

5.3.1 翻訳確率の問題点

提案手法の誤り解析から翻訳確率の問題点を考察する.

5.3.1.1 誤出力の結果

解析する提案手法の誤出力の結果を表 5.13 に示す．また，翻訳確率を表 5.14 に示す. この出力結果は複数作成された出力候補文の中で翻訳確率が第 1 位である.

表 5.13: 提案手法の誤出力の結果

入力文	その翌日のバスの切符を買った。
参照文	He booked himself for the following day's bus.
文パターン (日本語側)	*X3 X0 X4 X5 X1 X2* た。
文パターン (英語側)	*X3 X2 X1 X4 X0 X5* .
変換テーブル $_{CD}(X0)$	*C* : 翌日　　*D* : next
変換テーブル $_{CD}(X1)$	*C* : の切符　*D* : ticket for
変換テーブル $_{CD}(X2)$	*C* : を買っ　*D* : bought a
変換テーブル $_{CD}(X3)$	*C* : その　　*D* : I
変換テーブル $_{CD}(X4)$	*C* : の　　*D* : the
変換テーブル $_{CD}(X5)$	*C* : バス　　*D* : bus
出力文	I bought a ticket for the next bus.

表 5.14: 出力文の翻訳確率

出力候補文	I bought a ticket for the next bus.
$logP_v$	-48.1586
$logP_p$	-3.5850
$logP_m$	-2042.5963
$logP$	-2094.3399

表 5.13 の出力文からは「翌日」を表す英語句が存在しない. また，表 5.14 を見ると，言語モデルの確率 ($logP_m$) が，他の 2 つの確率 ($logP_v$ と $logP_p$) と比較して非常に小さい.

5.3.1.2 比較的正しい出力候補文の結果

比較的正しい出力候補文「I bought a ticket for bus in the next day.」が作成されていた．また，複数作成された翻訳候補文の中で翻訳確率が第 3 位である．文の作成結果を表 5.15 に示す．また，翻訳確率を表 5.16 に示す．

表 5.15: 出力候補文の作成結果

文パターン (日本語側)	その *X2 X0 X4 X1 X3* た 。
文パターン (英語側)	I *X3 X1 X4 X0* the *X2* .
変換テーブル $_{CD}(X0)$	*C* : の　　*D* : in
変換テーブル $_{CD}(X1)$	*C* : の 切符　*D* : ticket for
変換テーブル $_{CD}(X2)$	*C* : 翌日　　*D* : next day
変換テーブル $_{CD}(X3)$	*C* : を 買っ　*D* : bought a
変換テーブル $_{CD}(X4)$	*C* : バス　　*D* : bus
出力候補文	I bought a ticket for bus in the next day.

表 5.16: 出力候補文の翻訳確率

出力候補文	I bought a ticket for bus in the next day.
$logP_v$	-27.9219
$logP_p$	-3.9069
$logP_m$	-2290.6281
$logP$	-2322.4569

表 5.15 の出力候補文は翻訳確率 ($logP$) が表 5.13 の出力文と比較して小さかったため，出力文として選択されなかった．また，P_v は比較的正しい出力候補文 (表 5.16) の方が，誤出力文 (表 5.14) より大きい．

5.3.1.3 解析

表 5.14 と 5.16 から以下の 2 点が分かる．

- P_m は P_v と P_p より非常に小さい．
- 比較的正しい出力候補文の P_v の方が，誤った出力文の P_v より大きい．

従って，比較的正しい出力候補文を選択する方法として 2 つの方法がある．

1. 別の種類の言語モデルを使用する．
2. P_v，P_p，P_m に重みを付ける．

5.3.2 誤った変換テーブルについて

提案手法の誤り解析から誤った変換テーブルの問題点を考察する.

5.3.2.1 誤出力の結果

解析する提案手法の誤出力の結果を表 5.17 に示す.

表 5.17: 提案手法の誤出力の結果

入力文	雨の日が続いた。
参照文	He booked himself for the following day's bus.
文パターン (日本語側)	*X2 X0 X1* が *X3* た。
文パターン (英語側)	*X2* has *X3 X0 X1*.
変換テーブル $_{CD}(X0)$	*C* : の　　*D* : in the
変換テーブル $_{CD}(X1)$	*C* : 日　*D* : day
変換テーブル $_{CD}(X2)$	*C* : 雨　*D* : It
変換テーブル $_{CD}(X3)$	*C* : 続い　　*D* : been
出力文	It has been in the day.

表 5.17 の出力文からは「雨」を表す英語句が存在しない. また，変換テーブル $_{CD}(X2)$ を見ると，明らかに *CD* の意味が異なっている.

5.3.2.2　誤った変換テーブルの作成例

表 5.17 の下線部の変換テーブルが作成された手順を表 5.18 に示す.

表 5.18: 誤った変換テーブルの作成手順

学習文対 (日本語側)	物価が上がる。
学習文対 (英語側)	Prices rise.
単語レベル文パターン (日本語側)	*X1* が *X0*。
単語レベル文パターン (英語側)	*X1 N0* .
照合する学習文対 (日本語側)	雨が激しく降る。
照合する学習文対 (英語側)	It rains violently.
変換テーブル *(X0)*	*A*:上がる　*B*:rise *C*:激しく降る　*D*:rains violently
変換テーブル *(X1)*	*A*:物価　*B*:Price *C*:雨　*D*:It

下線部の変換テーブル *X1* が誤った変換テーブルである．なぜ，この変換テーブルが作成されたのか考察する.

5.3.2.3 解析

表 5.18 から以下の点が分かる．

- 照合する学習文対の日本語側の名詞「雨」と，英語側の名詞「It」が意味的に不一致である．

英語単語「It」は日本語文において翻訳されることが少ない．よって，変換テーブルとして抽出した際に，問題を発生させると考えられる．従って，正しい変換テーブルを作成する方法として 2 つの方法がある．

1. 単語レベル文パターン作成時に「It」を変数にしない．
2. 変換テーブル作成時に「It」を含めない．

今後，本節で考えた手法に取り組んでいきたい．

5.3.3 誤りの種類毎の誤出力数

本節では，TDPBSMT の誤出力における「翻訳確率の問題による誤出力」と「誤った変換テーブルによる誤出力」の総数を調査する．その結果から，どちらの問題により取り組むべきかを考察する．また，調査対象は TDSMT との対比較評価で TDSMT ○の評価かつ，明らかに誤りと判断できた 9 文である．調査した結果を表 5.19 に示す．

表 5.19: 誤りの種類毎の誤出力数

翻訳確率の問題による誤出力	7 文
誤った変換テーブルによる誤出力	2 文

出力結果を解析すると「誤った変換テーブルによる誤出力」がされた入力文でも, 翻訳確率 2 位以下に正しい翻訳候補分が存在する場合が多く.「翻訳確率の問題による誤出力」が多いことが分かった. よって, 今後は「翻訳確率の問題による誤出力」に対してより取り組むべきであると考える.

第6章　おわりに

本研究では，従来手法のカバー率の向上を目的として，文パターンを使用して翻訳を行う，“相対的意味論に基づく変換主導型パターンベース統計機械翻訳” を提案した．実験結果より，提案手法によって，カバー率が向上した．また，翻訳精度は従来手法と同等であった．以上より，提案手法は従来手法と比較して有効であると言える．今後は，提案手法の翻訳精度を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．(予定)

[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.