概要

近年，文法構造が大きく異なる言語間での翻訳において階層型統計翻訳が注目されている．

先行研究として，後藤ら [1] は特許文を用いて階層型統計翻訳と，句に基づく統計翻訳の自動評価・人手評価のスコア比較を行った．

その研究において，日英間の翻訳は階層型統計翻訳の人手評価が高く，英日間の翻訳は句に基づく統計翻訳の自動評価が高い結果となった．しかし，特許情報文は文の構造が複雑なため，翻訳結果の解析まで行われていない．

そこで，本論文では日本語-英語翻訳において，特許情報文と比べ比較的に人手評価の容易な，単文と重文複文を用いて，階層型統計翻訳と句に基づく統計機械翻訳の自動評価と人手評価を行った．その結果，階層型統計翻訳が両者の評価で高い結果となった．この原因として，階層型統計翻訳は階層を用いているので，文の構造が考慮されているが，句に基づく統計翻訳は句の並び替えによって，翻訳を行うので文の構造が考慮されていないことが考えられる．

そこで両者の評価結果から，翻訳方法の違いに着目し，主語と述語が翻訳されている文について再度人手調査を行った．調査の結果，階層型統計翻訳は 25 文，句に基づく統計翻訳は 12 文と句に基づく統計翻訳より階層型統計翻訳の文数が 13 文多い結果となった．この調査の結果，階層型統計翻訳は文の構造が考慮されることで翻訳評価が高くなったと考えられる．

目 次

図 目 次

表 目 次

第1章　はじめに

機械翻訳の歴史は文法規則や変換規則などを用いて翻訳を行うルールベース翻訳から始まる．そして 1960 年代半ばに，大量の翻訳対から作成した文パターン辞書を用いて翻訳を行うパターン翻訳が提案される．パターン辞書は人手で作成するので，開発に時間がかかる [2] が，文パターンに適合した場合に翻訳精度の高い翻訳文が得られる．1990 年代前半に「語に基づく統計翻訳」が提案された．初期の統計翻訳は，語に基づく翻訳モデルを用いていた．語に基づく翻訳モデルでは，単語の対応作成時に，対応が無い単語には NULL を対応させる．しかし，双方向の対応を調べる時，NULL に対する翻訳候補には，全ての単語が挙げられる．このことが，語に基づく翻訳モデルにおいて翻訳精度が低下する原因の一つになっていた．

しかし，2000 年の初めに「句に基づく統計翻訳 [3]」が提案され，「単語に基づく統計翻訳」と比べて翻訳精度が高いことから，現在，機械翻訳において統計翻訳が主流となっている．2005 年,「階層型統計翻訳 [5]」が提案され，文法構造が大きく異なる言語間での翻訳においての翻訳精度が期待されている．

2011 年，後藤らの研究により様々な翻訳手法において翻訳結果の自動評価と人手評価が行われた．その研究で，特許文を用いて階層型統計翻訳と句に基づく統計翻訳の評価を行った．日本語-英語間の翻訳の自動評価は階層型統計翻訳が高く，英語-日本語間の翻訳の人手評価は句に基づく統計翻訳が高くなった．しかし，階層型統計翻訳と句に基づく統計翻訳の性能の差を調査するための人手解析までは行われていない．理由として解析が困難な特許文を使用している点が挙げられる．

そこで本研究では，比較的容易に解析可能な単文・重文複文を用いて翻訳を行い翻訳結果の解析を行った．その結果，単文・重文複文において句に基づく統計翻訳より階層型統計翻訳が自動評価と人手評価共に高いスコアが出た．この原因として，句に基づく統計翻訳は語の並びによって翻訳するのに対し，階層型統計翻訳は階層的に翻訳を行うため，文の構造が考慮されているのではないかと考えた．よって，翻訳出力が，主語と述語が翻訳されているか調査した．その結果，句に基づく統計翻訳が 12 文，階層型統計翻訳は 25 文だった．結果より，階層型統計翻訳が文の構造を考慮しているため，翻訳精

度が高くなったと考えられる．

本論文の構成は以下の通りである．2章で統計翻訳システムの概要を説明する．3章で評価方法について説明する．4章で実験について説明する．5章で実験結果について説明する．6章で考察し7章でまとめる．

第2章　日英統計翻訳システム

2.1　句に基づく統計翻訳システム

句に基づく統計翻訳とは，翻訳する言語と目的言語の対訳文を大量に収集した対訳データを用いて，自動的に翻訳規則を獲得し翻訳を行う，機械翻訳手法の 1 つである．以下に，句に基づく統計翻訳の例を示す.

日英統計翻訳は，日本語文 J が与えられたとき，全ての組み合わせから確率が最大となる英語文 E を探索し翻訳を行う．翻訳モデルは $p(j|e)$，言語モデルは $p(e)$ である. 図 2.1 に句に基づく統計翻訳の手順を示す.

$$E = \arg\max_{e} P(e|j) \approx \arg\max_{e} P(j|e)P(e)$$

- 手順 1：日英対訳学習文より翻訳モデルを学習
- 手順 2：日英対訳学習文の英語文より言語モデルを学習
- 手順 3：手順 1，2 で学習した翻訳モデルと言語モデルを用いて翻訳

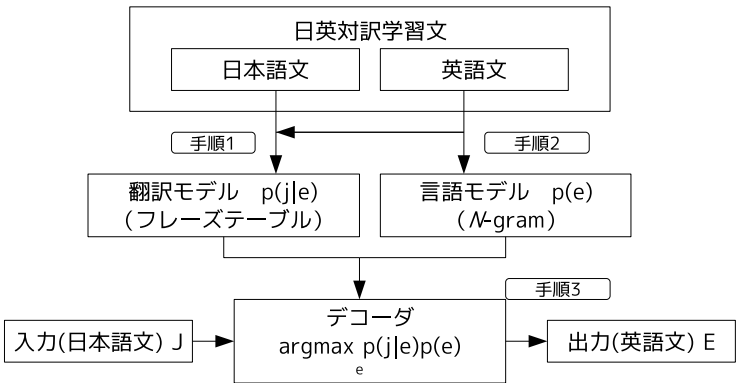

図 2.1: 句に基づく統計翻訳の手順

図 2.1 で示すように，翻訳モデルは日本語コーパスと英語コーパスが集まった，学習データから学習して作成する．また，言語モデルは，出力文の言語である英語コーパスから学習して作成する．翻訳モデルと言語モデルを用いて，E を探索する翻訳システムが図中のデコーダである．

2.1.1 翻訳モデル

翻訳モデルは，英語の単語の列から単語の列へ確率的に翻訳を行うためのモデルである. 翻訳モデルには，主に単語に基づくモデルと句に基づくモデルがある. 現在は，訳語の選択能力や局所的な語の並べ替え能力の高い，句に基づく翻訳モデルが現在の主流になっている.
翻訳モデルはフレーズテーブルで管理されている. 句に基づく統計翻訳は，フレーズテーブルという表で管理されている. 図 2.1 に例を示す.

表 2.1: フレーズテーブルの例

きみ ||| you ||| 0.05 0.0128243 0.2 0.000559831
ここ ||| here ||| 0.333333 0.150746 0.111111 0.186309
この ||| This ||| 0.000267023 0.0112064 0.00729927 0.414686
テレビ や ||| Television and ||| 0.25 0.00150427 1 0.0269209
テレビ や 新聞 ||| Television and newspapers ||| 0.125 0.00114611 1

左から，日本語フレーズ，英語フレーズ，フレーズの日英翻訳確率，単語の日英翻訳確率の積，フレーズの英日翻訳確率，単語の英日翻訳確率の積である.

2.1.2 IBM 翻訳モデル

翻訳モデルの代表例として，Brown らが提案した IBM の仏英翻訳モデル [6] がある．このモデルは，順に複雑な計算を行うモデル 1 からモデル 5 の 5 つのモデルから成る．IBM 翻訳モデルではフランス語から英語への翻訳を想定しているため，フランス語を F，英語文を E として説明を行う．IBM モデルでは仏語文 F，英語文 E の翻訳モデル $P(F|E)$ を計算するためにアライメント a と呼ばれる概念を導入し，以下のような式を考える．

$$P(j|e) = \sum_{a} P(j, a|e) \tag{2.1}$$

アライメントとはある仏単語 F と英単語 E の対応関係を意味している．IBM モデルのアライメントでは，仏英翻訳の場合，各英単語 e に対応する仏単語は 1 対 n の対応を持ち，仏語の単語は 1 つの英単語のみと対応すると仮定する．また，仏語の単語 f の対応関係として適切な英単語がなかった場合，英語文の文頭の特殊文字 e_0 と対応付けを行う．

2.1.2.1 モデル 1

式 (2.1) は以下の式に置き換えられる．

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

(2.2) 式は複雑なため計算が困難である．そこで，モデル 1 では以下の過程により，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．

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

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

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

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，$1 \leqq s \leqq S$)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．

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

手順3

英語文$E_{(s)}$のなかで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.(2.6)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．

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

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．

2.1.2.2 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 番目の仏英語 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．

$$p(F|E) = \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \Pi_{j=1}^{m} t(f_i|e_{a_j}) a(a_j|j,m,l) \tag{2.10}$$

$$= \epsilon \Pi_{j=1}^{m} \sum_{i=0}^{l} t(f_i|e_{a_j}) a(a_j|f,m,l) \tag{2.11}$$

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．

$$c(f|e;F,E) = \frac{t(f|e)}{t(f|e_0)+\cdots+t(f|e_l)} \sum_{j=1}^{m} \delta(f,f_j) \sum_{i=1}^{l} \delta(e,e_i) \tag{2.12}$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{l} \frac{t(f|e)a(i|j,m,l)\delta(f,f_j)\delta(e,e_i)}{t(f|e_0)a(0|j,m,l)+\cdots+t(f|e_l)a(l|j,m,l)} \tag{2.13}$$

$$c(i|j,m,l;F,E) = \sum_{a} P(a|E,F)\delta(i,a_j) \tag{2.14}$$

$$= \frac{t(f_j|e_i)a(i|j,m,l)}{t(f_j|e_0)a(0|j,m,l)+\cdots+t(f_j|e_l)a(l|j,m,l)} \tag{2.15}$$

$c(f|e;F,E)$ 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を表し，$c(i|j,m,l;F,E)$ 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 は，複数の極大値を持つため，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．モデル 1 は最適解に必ず収束するため，モデル 1 を用いることで最適解を得ることができる．

2.1.2.3　モデル 3

モデル 3 は，英単語と仏単語の対応は 1 対 1 の場合のみを想定していたモデル 1，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 3 つ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．

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

以上により，モデル 3 は以下の式で求められる．

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

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．

2.1.2.4　モデル 4

モデル 4 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

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

$\odot_{i_1}$ 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

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

$\pi_{[i]k-1}$ 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．

2.1.2.5　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

$$\begin{aligned} P(\Pi_{[i]1} \ &= \ j|\pi_1^{[i]-1},\tau_0^l,\phi_0^l,E) \\ &= \ d_1(v_j|\mathcal{B}(f_j),v_{\odot_{i-1}},v_m-\phi_{[i]}+1)(1-\delta(v_j,v_{j-1})) \end{aligned} \tag{2.21}$$

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

$$\begin{aligned} P(\Pi_{[i]k} \ &= \ j|\pi_{[i]1}^{k-1},\pi_1^{[i]-1},\tau_0^l,\phi_0^l,E) \\ &= \ d_{>1}(v_j-v_{\pi_{[i]k-1}}|\mathcal{B}(f_j),v_m-v_{\pi_{[i]k-1}}-\phi_{[i]}+k)(1-\delta(v_j,v_{j-1})) \end{aligned} \tag{2.22}$$

2.1.3 GIZA++

GIZA++[4] とは，統計翻訳で用いることを前提に作られた単語対応のアライメントを行うツールである．IBM モデル 1〜5 を学習し，単語の対応関係の確率値を計算する．

2.1.4　フレーズテーブルの作成法

はじめに，GIZA++を用いて学習文から日英，英日方向の双方向で最尤な単語アライメントを得る．

日英方向の単語対応の例を表に示す．英日方向の単語対応の例は表 2.2 に示す．表中の“●” は得られた単語アライメントを示す．

表 2.2: 日英方向の単語対応

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●	●							
took									●	
a										
drive							●	●		
on						●				
the										
stock				●						
market					●					
.										●

表 2.3: 英日方向の単語対応

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●								
took							●			
a							●			
drive							●			
on							●			
the						●				
stock				●						
market					●					
.										●

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 2.2 と表 2.3 を用いた，“intersection” での例を表 2.4，“union” での例を表 2.5 に示す．

表 2.4: intersection の例

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●								
took										
a										
drive							●			
on										
the										
stock				●						
market					●					
.										●

表 2.5: union の例

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●	●							
took							●		●	
a							●			
drive							●	●		
on						●	●			
the						●				
stock				●						
market					●					
.										●

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” の例を表 2.6，“grow-diag” の例を表 2.7 に示す．

表 2.6: grow の例

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●	●							
took										
a							●			
drive							●	●		
on							●			
the										
stock				●						
market					●					
.										●

表 2.7: grow-diag の例

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●								
took										
a										
drive							●			
on						●				
the										
stock				●						
market					●					
.										●

また，この “grow” と “grow-diag” の最後に行う処理として “final” と “final-and” がある．“final” は “union” の単語対応があれば用いる．“final-and” では，“final” に加えて，双方向ともに単語対応がないアライメントも用いる．“grow-diag-final” の例を表 2.8，“grow-diag-final-and” の例を表 2.9 に示す．

表 2.8: grow-diag-final の例

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●	●							
took							●		●	
a							●			
drive							●	●		
on						●	●			
the						●				
stock				●						
market					●					
.										●

表 2.9: grow-diag-final-and の例

	鉄道	株	が	株式	市場	で	暴落	し	た	。
Rail	●									
stocks		●	●							
took							●			
a							●			
drive							●	●		
on						●	●			
the						●				
stock				●						
market					●					
.										●

そして，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．“grow-diag-final-and” で作成されたフレーズテーブルの例を表 2.10 に示す．

表 2.10: grow-diag-final-and で作成されたフレーズテーブルの例

鉄道 ||| Rail
株 が ||| stocks
鉄道 株 が ||| Rail stocks
市場 ||| market
で 暴落 し た ||| took a dive on the
た 。 ||| .

2.1.5 言語モデル

言語モデルは単語列の生じる確率を与えるモデルである．日英翻訳では，翻訳モデルで生成された翻訳候補から英語として自然な文を選出する．統計翻訳では一般に，N-gram モデルを用いる．

表 2.11: 言語モデルの例

-0.9121773 factory . -0.772665
-1.571392 factory has -0.05683998
-1.120353 factory in -0.05121826
-1.821027 factory will -0.0660101
-1.56243 facts do -0.2219447
-1.232086 facts of -0.227057

一番上の行に関して，左から，“factory” のあとに “.” がくる確率を常用対数で表した値 “$log_10(P(a|jfactory)) = -0.9121773$”，2-gram で表された単語列である “factory .”，バックオフスムージングにより得られる，“factory” のあとに “.” がくる確率を常用対数で表した値 “$log_10(P(a|jfactory)) = -0.772665$” である．また，バックオフスムージングとは，高次の N-gram が存在しない場合，低次の N-gram を用いる手法である．この低次の確率を，改良したスムージングの手法が Kneser-Ney スムージングである．言語モデルにおける N-gram 作成には，性能の観点から一般的に Kneser-Ney スムージングが用いられている．

2.1.6 *N*-gram モデル

代表的な言語モデルに N-gram がある．*N*-gram は「単語の列 $(w_1^i = w_1, w_2, \cdots, w_i)$ の i 番目の単語 (w_i) の生起確率 (Pw_i) は直前の $(i-1)$ 単語に依存す」，という仮説に基づくモデルである. 計算式を以下に示す.

$$P(w_1^i) = \prod_i^{k=1} P(w_i \mid w_{i-1})$$

例えば，「I am a teacher .」という文字列に対する 2-gram モデルを以下に示す．$P(e =$ “$I\ am\ a\ teacher\ .$”$) \approx P(I) \times P(am \mid I) \times P(a \mid am) \times P(teacher \mid a) \times P(. \mid teacher)$ また，3-gram モデルのときは $P(a \mid I\ am)$ になり，4-gram モデルのときは $P(teacher \mid I\ am\ a)$ になる．このように，(N-1) 単語の次にくる単語が “a” や “teacher” である確率を考える．

2.1.7 デコーダ

デコーダは翻訳モデルと言語モデルの全ての組合せから確率が最大となる出力文を探索して翻訳を行う．この探索には莫大な計算量が必要となるが，ビームサーチ法を用いて候補をしぼることで計算量を減らす．代表的なデコーダとして “Moses[7]” や “Joshua[8]” がある．

2.1.7.1 moses のパラメータ

moses で設定できるパラメータの例を以下に示す.

weight-l:言語モデルの重み

weight-t:翻訳モデルの重み

weight-d:リオーダリングの重み

weight-w:目的言語の長さに関するペナルティー

distotion-limit:フレーズ並び替えの制限範囲

2.1.7.2　ビームサーチ法

ビームサーチ法は，探索の計算量を減らすために用いられる．ビームサーチ法は，翻訳候補の探索木において，翻訳確率の低い翻訳候補を枝刈りし，探索の範囲を限定する．枝刈りは “histogram pruning” と “threshold pruning” によって行う．“histogram pruning” は確率の高い翻訳候補のみを一定数残す枝刈り法である．“threshold pruning” は一定の確率以上の翻訳候補のみを残す枝刈り法である．この 2 つの枝刈り法を用いて，探索範囲を限定する．しかし，翻訳が進むほど，翻訳候補の確率は小さくなる．そのため，翻訳が進んだ翻訳候補と翻訳が進んでいない翻訳候補を比較したとき，翻訳が進んだ翻訳候補ほど枝刈りの対象となる可能性が高い．

2.1.7.3　パラメータチューニング

パラメータチューニングには，MERT（Minimam Error Rate Training）[9] が用いられる．MERT は，目的の自動評価（一般的に BLEU）を最大にするような翻訳結果を出力するようにパラメータを調整する．その際ディベロップメントデータと呼ばれる，試し翻訳を行うデータを使用し，各文に対し上位 100 文程度の翻訳候補を出力する．その候補の中で重みを変えることでよりよい翻訳候補が上位にくるようにパラメータを調整する．

2.2 階層型統計翻訳システム

階層型統計翻訳とは，句に基づく統計翻訳と同様の翻訳する言語と目的言語の対訳文を大量に収集した対訳データを用いて，自動的に翻訳規則を獲得し翻訳を行う，機械翻訳手法の 1 つである．階層型統計翻訳の特徴として，翻訳モデルにルールテーブルを用いる．また，デコーディングの際には木構造を用いる．以下に，階層型統計翻訳の例を示す.

階層型統計翻訳は，日本語文 J が与えられたとき，全ての組み合わせから確率が最大となる英語文 E を探索し翻訳を行う．翻訳モデルは $p(j|e)$，言語モデルは $p(e)$ である. 図 2.2 に階層型統計翻訳の手順を示す.

$$E = \arg\max_{e} P(e|j) \approx \arg\max_{e} P(j|e)P(e)$$

- 手順 1：日英対訳学習文より翻訳モデルを学習
- 手順 2：日英対訳学習文の英語文より言語モデルを学習
- 手順 3：手順 1，2 で学習した翻訳モデルと言語モデルを用いて翻訳を行う.

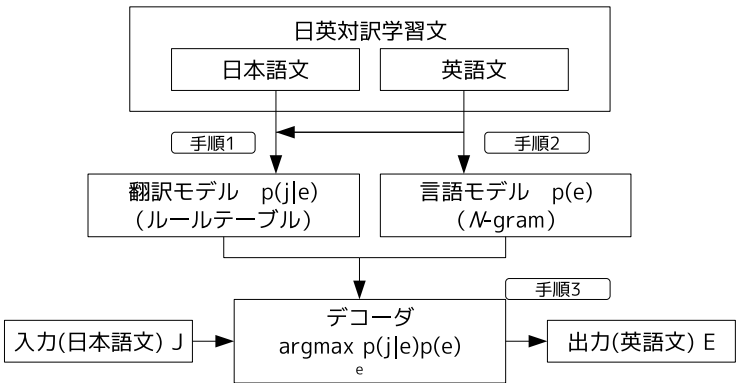

図 2.2: 階層型統計翻訳の手順

2.3 階層型統計翻訳の概要

句に基づく統計翻訳がフレーズ毎に翻訳を評価するモデルであったのに対して，その句を階層にすることで構文単位で評価するモデルにしたのが階層型統計翻訳である．階層型統計翻訳の例を図 2.3 に示す．この例のように，句のペアにそれぞれ穴を空け，この穴の中に別のフレーズペアを埋め込むことができるようにする．このような句のペアを考えると図 2.4 のような翻訳が可能となる．

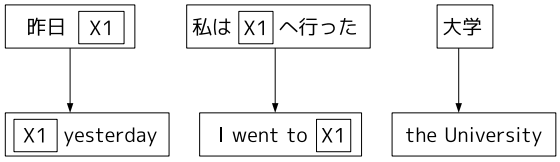

図 2.3: 階層フレーズの例

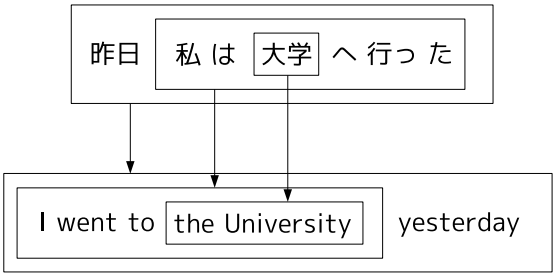

図 2.4: 階層ルールによる翻訳例

この例のように，フレーズの中にフレーズを埋め込むようにして，対訳文対の翻訳確率を評価する．

2.3.1 翻訳モデル

2.3.1.1 ルールテーブル作成法 (ルールの抽出)

階層型統計翻訳は句に基づく統計翻訳を階層に拡張したものであり，ヒューリスティクスによる推定はほぼ階層でない句に基づく統計翻訳と同じである．2.3.1 節では句に基づく統計翻訳と異なる点を説明する．

まず，ルールの抽出について述べる．階層型統計翻訳においても，学習データに表われるルールを全て列挙することは現実的でない．そこでルールを全て列挙することは行わず，ルールとして正しそうなもののみを抽出することになる．まず，階層でないフレーズモデルにおいてフレーズペアを抽出する．これで得られるフレーズペアの集合には，フレーズペアとして同じ単語アラインメントを持っているものが複数あるため，フレーズペアとして最小であるものを選ぶ．これによって得られたフレーズペア集合を初期フレーズペア (initial phrasepair) 集合と呼ぶ．初期フレーズペアが得られると，以下の定義に従ってルールを得る．

- 初期フレーズペアはルールである
- あるルール $r_1 = (f_1, e_1)$ と別のルール $r_2 = (f_2, e_2)$ があり，$f1 = \overline{f_1'}, \overline{f_2 f_1''} \wedge e_1 = \overline{e_1' e_2 e_1''}$ で表わされるなら，$r^{new} = (\overline{f_1'} X \overline{f_1''}, \overline{e_1'} X \overline{e_1''})$ はルールである

この抽出を各文のフレーズについて行い，ルールの集合を得る．

2.3.1.2 ルールテーブル作成法 (ルールの確率推定)

階層型統計翻訳のルールの確率が階層型統計翻訳の feature として含まれているため各ルールの確率を計算しなければならない．この値もヒューリスティクスで与える．このヒューリスティクスも句に基づく統計翻訳と同じであるが，階層型統計翻訳においては一つの初期フレーズペアから複数のルールが抽出されるため，初期フレーズペアから抽出されたフレーズペアのカウントが足して 1 となるように一様に分散させ，そのカウントを元にフレーズモデルと同様に確率を与える．

2.3.2 デコーダ

デコーダにおいても基本的には句に基づく統計翻訳と同様に $argmax_e P(e|f)$ を計算するためのシステムである．翻訳モデルと言語モデルのルールを用いて確率が最大となる出力文の翻訳を行う．この翻訳にも莫大な計算量が必要となり，Cube Pruning 法 [10] を用いて候補をしぼることで計算量を減らす．代表的なデコーダとして “Moses” や “Joshua” がある．

2.3.3 Cube Pruning

Cube Pruning 法は，翻訳の探索空間を狭める手法である．Cube Pruning 法は，A* 探索 [10] と本質的には同じであり，Cube pruning は最適解を求める保証がないヒューリスティックだが，A* 探索として再定式化すると，厳密解が求められなかったところを厳密解を求めることができ，計算時間をほとんど変えずに翻訳の精度を上げることができる．

第3章　評価方法

3.1　人手評価

人手評価には，対比較評価を用いる．評価区分を以下に示す．

○ **PSMT** ：PSMT の翻訳結果が優れている場合

○ **HSMT** ：HSMT の翻訳結果が優れている場合

差なし ：PSMT と HSMT の翻訳出力の質に差が無い場合

同一出力 ：PSMT と HSMT の出力が同一である場合

3.2　自動評価

機械翻訳システムの翻訳精度を自動的に評価する手法として，あらかじめ用意した正解文と，翻訳システムが出力した文とを比較する手法が一般的である．自動評価法には多くの方法がある．本研究では，BLEU[13] と NIST[14] と METEOR[15] と IMPACT[16] と RIBES[17] と TER[18] と WER[18] を用いる．BLEU は語順 (4-gram) が正しい場合に高いスコアを出す．NIST では BLEU と同様に語順の正しさで比較を行うが，5-gram を用いる．METEOR は単語属性 (3 人称単数など) が正しい場合に高いスコアを出す．IMPACT は，名詞句の塊が正しく配置されている場合に高いスコアを出す．RIBES は，文全体の大局的な並びが正しい場合に高いスコアを出す．TER(Translation Error Rate) は，翻訳結果から正解文に変換する手順を調べ，翻訳誤りの割合を出す．WER(Word Error Rate) は，単語が正しく変換されているか調べて，単語誤りの割合を出す．BLEU と METEOR と IMPACT と RIBES と TER と WER では 0 から 1 までの間で評価され，NIST では 0 から∞までの間で評価される．BLEU と NIST と METEOR と IMPACT と RIBES の評価方法は，評価方法が高いほど翻訳精度が高いことを表す．TER と WER の評価方法は，評価方法が低いほど翻訳精度が高いことを表す．尚，本研究では入力文 1 文に対して正解文 1 文を用いて評価を行う．

3.2.1 BLEU

BLEU は，機械翻訳システムの自動評価において，現在主流になっている評価法である．BLEU は，*N*-gram 適合率で比較を行う．実験では，4-gram を用いる．BLEU は，0 から 1 のスコアを出力し，スコアが 1 に近いほど良い評価である．BLEU の計算式を式 3.1 に示す．

$$BLEU = BPexpW_n \sum_{n=1}^{N} (\log_e P_n) \tag{3.1}$$

$$W_n = \frac{1}{N} \tag{3.2}$$

$$P_n = \frac{\sum_i \text{出力文中 } i \text{ と参照文 } i \text{ で一致した } N-gram \text{ 数}}{\sum_i \text{出力文中 } i \text{ の中の全 } N-gram \text{ 数}} \tag{3.3}$$

ここで，BP は短い翻訳文が高い評価にならないように補正を行うパラメータである．また，W_n は *N*-gram の重みである．

3.2.2 NIST

NISTは，BLEUと同様に*N*-gram適合率で評価を行う．情報量で重み付けをしている点が異なる．また，本実験では5-gramを用いる．NISTは，0から∞のスコアを出力し，スコアが大きいほど良い評価となる．NIST計算式を式3.4に示す．

$$NIST = \sum_{n-1}^{n} \frac{\sum_i (\sum_{\text{出力文 } i \text{ と参照文 } i \text{ に共通する } w_i \cdots w_n} INFO(w_i \cdots w_n))}{\sum_i \text{出力文 } i \text{ の中の全 } N-gram \text{ 数}} \quad (3.4)$$

$$INFO(w_i \cdots w_n) = \log_2 \frac{\text{評価コーパス中の } w_i \cdots w_{n-1} \text{数}}{\text{評価コーパス中の } w_1 \cdots w_n \text{数}} \quad (3.5)$$

3.2.3 METEOR

METEORは，単語属性が正しい場合に高いスコアを出す. 実験では2-gramを用いる．METEORは0から1までのスコアを出力し，スコアが1に近いほど良い評価となる．計算式を式3.8に示す．

$$F\text{値} = \frac{P \times R}{\alpha \times P + (1-\alpha) \times R} \quad (3.6)$$

$$Pen = \gamma \times (\frac{c}{m})^{\beta} \quad (3.7)$$

$$METEOR = F \times (1 - Pen) \quad (3.8)$$

METEORはF値，ペナルティ関数Penを用いて計算される．F値は適合率Pと再現率Rの調和平均で求められる．そしてペナルティ関数Penにおいて，mは参照文と出力文の間で一致した単語数を示す．またcは，一致した単語を対象として，参照文と一致する単語列を1つのまとまりの数を示す．したがって，参照文と出力文が同一文である場合はc=1となる．尚，α,β,γの値はパラメータである．

3.2.4 IMPACT

IMPACT は，名詞句のかたまりを用いて評価を行う手法である．参照文と出力文において，対応する名詞句を用いて，一致する単語列を正確に決定する．さらに，名詞句の出現順に関して類似性を決定する．IMACT は再現率と適合率から F 値を求め，F 値を IMPACT のスコアとしている．計算式を式 3.11 に示す．

$$R = \left(\frac{\sum_{i=1}^{RN} (\alpha^i \sum_{c \in LCS} length(c)^\beta)}{m^\beta} \right)^{\frac{1}{\beta}} \tag{3.9}$$

$$R = \left(\frac{\sum_{i=1}^{RN} (\alpha^i \sum_{c \in LCS} length(c)^\beta)}{m^\beta} \right)^{\frac{1}{\beta}} \tag{3.10}$$

$$score = \frac{(1+\gamma^2)RP}{R+\gamma^2 P} \tag{3.11}$$

$$\gamma = \frac{P}{R} \tag{3.12}$$

ここで，LCS とは最長共通部分列であり，RN は LCS の決定プロセスの繰り返し数を示す．そして，i は RN に対するカウンターで，α と β はパラメータである．また m は参照文の単語数，n は出力文の単語数，length(c) は共通部分の単語数を示す．尚，IMPACT はスコアが大きい方が良い評価である．

3.2.5 RIBES

RIBES は，参照文と出力文との間で，共通単語の出現順序を順位相関係数で評価を行う評価法である．計算式を 3.13 と 3.14 に示す．

$$RIBES = NSR \times P^\alpha \tag{3.13}$$

$$RIBES = NKT \times P^\alpha \tag{3.14}$$

ここで，NSR はスピアマンの順位相関係数であり，NKT は，ケンドールの順位相関係数である．また α はペナルティに対する重みとして使用され，$0 \leqq \alpha \leqq 1$ の値である．単語の出現順を順位相関係数を用いて評価することで，文全体の語順に着目することができる．尚，RIBES は 0 から 1 のスコアを出力し，1 に近い方が良い評価である．

3.2.6 TER

TER は，Translation Error Rate の略で翻訳の誤り率を求める手法である．計算式を式 3.15 に示す．

$$TER = \frac{\sum_i (\text{挿入語数}\, i + \text{置換語数}\, i + \text{削除語数}\, i + \text{シフト語数}\, i)}{\sum_i (\text{参照文}\, i \,\text{の平均単語数})} \tag{3.15}$$

分子は，参照文と出力文の比較における編集操作数のことである.TER の編集操作は挿入，置換，削除，シフトの 4 種類の編集を行うことである．尚，TER はスコアが 0 に近いほど良い評価である．

3.2.7 WER

WER は，Word Error Rate の略で単語の誤り率を求める評価法である．計算式を式 3.16 に示す．

$$WER = \frac{\sum_i (\text{挿入語数}\, i + \text{置換語数}\, i + \text{削除語数}\, i)}{\sum_i (\text{参照文}\, i \,\text{の平均単語数})} \tag{3.16}$$

分子は，参照文と出力文の比較における編集操作数のことである．WER の編集操作は挿入，置換，削除の 3 種類の編集を行うことである．尚，WER はスコアが 0 に近い方がが良い評価である．

第4章　実験

4.1　実験環境

4.1.1　使用するコーパス

a. 単文コーパス

実験に，辞書の例文より抽出した単文コーパス 182,899 文 [11] から，学習データとして 100,000 文，テストデータとして 10,000 文，ディベロップメントデータとして 1,000 文を用いる．統計翻訳の前処理として，各コーパスの日本語文に対して，MeCab[19] を使用し形態素解析を行う．

b. 重文複文コーパス

実験には，辞書の例文より抽出した重文複文コーパス 122,719 文 [12] から，学習データとして 100,000 文，テストデータとして 10,000 文，ディベロップメントデータとして 1,000 文を用いる．統計翻訳の前処理は単文と同様の処理を行う．

4.2　実験内容

まずはじめに，単文と重文複文それぞれのコーパスを用いて，階層型統計翻訳と句に基づく統計翻訳の翻訳を行う．次に，単文コーパスを用いた階層型統計翻訳と句に基づく統計翻訳の自動評価を行う．自動評価は，3 章で紹介した自動評価方法 BLEU，NIST，METEOR，IMPACT，RIBES，TER，WER を使用する．最後に，階層型統計翻訳と句に基づく統計翻訳を人手で対比較評価する．重文複文も，同様にして翻訳から，自動評価と人手評価を行う．この自動評価と人手評価の結果より考察を行う．

第5章　実験結果

5.1　単文コーパスを用いた実験

5.1.1　PSMTの自動翻訳結果

PSMTの自動評価結果を表5.1に示す.

表 5.1: PSMTの自動評価結果

BLEU	NIST	METEOR	IMPACT	RIBES	TER	WER
0.1249	4.6376	0.4352	0.4285	0.6944	0.7197	0.7484

5.1.2　HSMTの自動評価結果

HSMTの自動評価結果を表5.2に示す.

表 5.2: HSMTの自動評価結果

BLEU	NIST	METEOR	IMPACT	RIBES	TER	WER
0.1372	4.8640	0.4595	0.4453	0.7148	0.7034	0.7304

5.1.3　人手評価

単文コーパスを用いたHSMTの人手評価結果を表5.3に示す.

表 5.3: 人手評価結果

PSMT	HSMT	差なし	同一出力
12	22	54	12

PSMT が 12 文，HSMT が 22 文，差なしが 54 文，同一出力が 12 文となり HSMT の文が多い結果となった．単文コーパスを用いた翻訳結果は，自動評価，人手評価ともに HSMT が高いスコアとなった．

5.1.4 PSMT の翻訳例

PSMT の翻訳結果が優れていると評価した例を表 5.4 に示す．

表 5.4: PSMT の翻訳結果が優れていると評価した例

翻訳例 1
入力文:星 が あらわれ はじめ た 。 正解文:The stars began to peep out . ○ PSMT:The stars began to あらわれ . × HSMT:The stars あらわれ began to .
翻訳例 2
入力文：我々は早く決定することに意見が一致した。 正解文：We agreed to decide quickly . ○ PSMT：We agreed on a quick decision . × HSMT：We are agreed to a quick decision .
翻訳例 3
入力文：両家 は 去年 縁組み を し た 。 正解文：The two families were united by marriage last year . ○ PSMT：The families 縁組み last year . × HSMT：縁組み families the last year .

翻訳例 1 について，PSMT は，動詞が正しい位置にあるため優れていると判断する．HSMT は，動詞が正しい位置にないため劣っていると判断する．

翻訳例 2 について，PSMT は，文法が正しく文の意味が通っているので優れていると判断する．HSMT は，文法が正しいが，受身型となり文の意味が変わっているため劣っていると判断する．

翻訳例 3 について，PSMT は，動詞が正しい位置にあるため優れていると判断する．HSMT は，動詞が正しい位置にないため劣っていると判断する．

5.1.5 HSMT の翻訳例

HSMT の翻訳結果が優れていると評価した例を表 5.5 に示す.

表 5.5: HSMT の翻訳結果が優れていると評価した例

翻訳例 1
入力文：彼は店の前に看板を立てる。 正解文：He will put up a sign in front of his store . × PSMT：He in front of the store or . ○ HSMT：He Set up a sign in front of the store .
翻訳例 2
入力文:彼女には文学の素養がある。 正解文:She has learned a good deal of literature. × PSMT:She has literary culture . ○ HSMT:There is a literary culture to her .
翻訳例 3
入力文:ふたり の 仲裁 に 入っ た 。 正解文:I mediated for the two . × PSMT:intervened between the two . ○ HSMT:He intervened between the two .

翻訳例 1 について，PSMT は，動詞が欠落し，文の意味が通っておらず劣っていると判断する．HSMT は，文法が正しく文の意味が通っているので優れていると判断する．

翻訳例 2 について，PSMT は，文法が正しく文の意味が通っているので良いと判断する．HSMT は，文法が正しく文の意味が通っているので優れていると判断する．

翻訳例 3 について，PSMT は，主語が翻訳されていないので劣っていると判断する．HSMT は，主語が翻訳されているので優れていると判断する．

5.1.6　翻訳結果に差が無い例

翻訳の質に差なしと判断した例を表 5.6 に示す.

表 5.6: 翻訳の質に差なしと判断した例

入力文：急に走るのは心臓に悪い。
正解文：It is bad for your heart to suddenly start running .
PSMT : I suddenly the heart in the wrong .
HSMT : I suddenly for the heart .

PSMT と HSMT のどちらも動詞が欠落し意味が通っておらず差なしと判断する.

5.1.7　翻訳結果が同一である例

同一出力の例を表 5.7 に示す.

表 5.7: 同一出力の例

入力文 ：彼の飛行機の出発時間を忘れた。
正解文 ：I am pledged to secrecy .
PSMT ：I pledged to keep the secret .
HSMT ：I pledged to keep the secret .

5.2 重文複文の自動評価結果

5.2.1 PSMT の自動評価結果

PSMT の自動評価結果を表 5.8 に示す.

表 5.8: PSMT の自動評価結果

BLEU	NIST	METEOR	IMPACT	RIBES	TER	WER
0.1032	4.2991	0.3985	0.3716	0.6425	0.8007	0.8458

5.2.2 HSMT の自動評価結果

HSMT の自動評価結果を表 5.9 に示す.

表 5.9: HSMT の自動評価結果

BLEU	NIST	METEOR	IMPACT	RIBES	TER	WER
0.1220	4.5702	0.4251	0.3972	0.6802	0.7658	0.8033

5.2.3 人手評価

HSMT の人手評価結果を表 5.10 に示す.

表 5.10: 人手評価結果

PSMT	HSMT	差なし	同一出力
7	25	65	3

重文複文コーパスを用いた翻訳結果は，自動評価，人手評価ともに HSMT が高いスコアとなった.

5.2.4 PSMT の翻訳例

PSMT の翻訳結果が優れていると評価した例を表 5.11 に示す.

表 5.11: PSMT の翻訳結果が優れていると評価した例

翻訳例 1
入力文:運命が彼にそこへ行くことを命じた。 正解文:Destiny decreed that he should be there . PSMT:He ordered a that we should go there to fate . HSMT:that fate that we should go there to him .
翻訳例 2
入力文 : 私は 旅行に出かけたくてたまらなかった。 正解文 : I was impatient to start on the trip . PSMT : I want to go on a trip . HSMT : I was dying I left on a trip .
翻訳例 3
入力文 :彼女 は 彼 の よい 妻 に なる だろ う 。 正解文 :She will make a good match for him . PSMT :She will be a good wife . HSMT :She he would be to be a good wife .

翻訳例 1 について，PSMT は，文法が正しく文の意味が通っているので優れていると判断する．HSMT は，命じるという部分が翻訳されておらず，文の意味が変わっているため劣っていると判断する．

翻訳例 2 について，PSMT は，文法が正しく文の意味が通っているので優れていると判断する．HSMT は，動詞が翻訳できておらず劣っていると判断する．

翻訳例 3 について，PSMT は，文法が正しく文の意味が通っているので優れていると判断する．HSMT は，主語が二つあるので劣っていると判断する．

5.2.5 HSMT の翻訳例

HSMT の翻訳結果が優れていると評価した例を表 5.12 に示す.

表 5.12: HSMT の翻訳結果が優れていると評価した例

翻訳例 1
入力文：英語に親しむように努めている。 正解文：I am trying hard to get more familiar with English . PSMT : began to drink habitually in English . HSMT : He tried to enjoy English .
翻訳例 2
入力文:法律 違反 者 を 逮捕 する 権限 を 与え られ て いる 。 正解文:I am invested with the authority to arrest lawbreakers . PSMT:an the authority to arrest . HSMT:He is an authorized to arrest the suspect .
翻訳例 3
入力文:その 失敗 は 我々 を 絶望 に おとしいれる ほど の もの で は なかっ た 。 正解文:The mistake was not such as to make us despair . PSMT:The failure us おとしいれる out of despair . HSMT:The failure was not enough to despair おとしいれる us .

翻訳例 1 について，PSMT は，主語が無く「努めている」に当たる動詞が無いため意味が通っておらず劣っていると判断する．HSMT は,「親しむ」が「楽しむ」と意味が変わっているがニュアンスが近い，また努めているという動詞を翻訳できているため優れていると判断する．

翻訳例 2 について，PSMT は，動詞が翻訳できておらず劣っていると判断する．HSMT は，文法が正しく文の意味が通っているので優れていると判断する．

翻訳例 3 について，PSMT は，動詞が翻訳されていないために劣っていると判断する．HSMT は，未知語以外は翻訳に問題がないので優れていると判断する．

5.2.6　翻訳結果に差が無い例

翻訳の質に差なしと判断した例を表 5.13 に示す.

表 5.13: 翻訳の質に差なしと判断した例

入力文 ：エイズ は アメリカ に 限ら れ た こと で は ない 。
正解文 ：AIDS was not confined to America .
PSMT ：AIDS is limited to the United States .
HSMT ：AIDS is limited in America .

PSMT と HSMT 共に否定型になっておらず翻訳できていない.

5.2.7　翻訳結果が同一である例

同一出力の例を表 5.14 に示す.

表 5.14: 同一出力の例

入力文：私は絶対に秘密を守ると誓っている。
正解文：I am pledged to secrecy .
PSMT : I pledged to keep the secret .
HSMT : I pledged to keep the secret .

第6章　考察

6.1　翻訳評価について

HSMT と PSMT では，単文において全ての自動評価と人手評価共に HSMT が高いスコアとなった．また，重文複文においても同様に，全ての自動評価と人手評価共に HSMT が高いスコアとなった．この原因として，句に基づく統計翻訳は語の並びによって翻訳するのに対し，階層型統計翻訳は階層的に翻訳を行うため，文の構造が考慮されているのではないかと考えた．そこで，翻訳出力が文の構造を考慮されているかを調査した．文の構造ができている基準として主語と述語が翻訳されているかを調査した．これは基本的な文が主語と述語で構成されているためである．調査対象は，重文複文における PSMT と HSMT の翻訳結果とする．この理由として人手評価と自動評価で大きな差が出たため，翻訳結果にも差が出ると考えたからである．表 6.1 に，結果を示す．

表 6.1: 主語と述語の評価結果

PSMT	HSMT
12	25

表 6.2 に示すように，PSMT よりも HSMT が主語と述語の翻訳ができている文数が多かった．よって，PSMT よりも HSMT が主語と述語の翻訳ができている文数が多いために自動翻訳・人手翻訳が高くなったと考える．

表 6.2: HSMT で主語と述語が翻訳された例

入力文:船は揺れるので嫌いです。
正解文:I do not like boats because they rock back and forth .
PSMT:The ship , so I hate shakes .
HSMT:The ship shakes , so I do not like .

尚，表 6.2 において，HSMT は日本語文の「船は揺れる」に対して，英語文の「The ship shakes」のように主語と述語の翻訳ができている．

第7章　おわりに

本論文では，単文・重文複文における階層型統計翻訳と句に基づく統計翻訳の翻訳結果を調査した．その結果，自動評価・人手評価共に句に基づく統計翻訳よりも階層型統計翻訳のほうが評価が高いスコアとなった．具体的な自動評価のスコアとして，単文においてそれぞれBLEUは0.0123，NISTは0.2264，METEORは0.0243，IMPACTは0.0172，RIBESは0.0204，TERは0.0163，WERは0.018，階層型統計翻訳のほうがスコアが高くなった．また重文複文においてBLEUは0.0188，NISTは0.2711，METEORは0.0266，IMPACTは0.0256，RIBESは0.0377，TER0.0349は，WERは0.0425，階層型統計翻訳のほうがスコアが高くなった．人手評価は，単文において句に基づく統計翻訳が12文，階層型統計翻訳が22文重文複文において句に基づく統計翻訳が優れていると評価したものが12文，階層型統計翻訳が優れていると評価したものが22文重文複文において句に基づく統計翻訳が優れていると評価したものが7文，階層型統計翻訳が優れていると評価したものが26文だった．単文・重文複文を使用した場合，自動評価と人手評価より階層型統計翻訳の評価が高くなることがわかった．

さらに，スコアが高くなった原因の調査として，文の構造を考慮するか，又はしないかのデコーディングの違いによって，評価に差が出たと考えた．よって，主語と述語の翻訳が出来ている文数を調査した．調査の結果，句に基づく統計翻訳が12文，階層型統計翻訳が25文翻訳ができていた．これより，HSMTは翻訳する際に文法構造が考慮されていて，主語と述語がよく翻訳されているためであると考える．

今後は，主語と述語以外の文法構造について調査していきたい．

第8章　謝辞

最後に，一年間に渡り，本研究の御指導をいただきました鳥取大学工学部知能情報工学科計算機講座 C 研究室の村田真樹教授，村上仁一准教授，徳久雅人講師に深く感謝するとともに厚くお礼を申し上げます

参考文献

[1] Taro Watanabe, Jun Suzuki, Hajime Tsukada and Hideki Isozaki. 2007.“Online Large-Margin Training for Statistical Machine Translation” In EMNLP-CoNLL pp. 764-773.(2007)

[2] 池原悟, 宮崎正弘, 白井諭, 横尾昭男, 中岩浩巳, 小倉健太郎, 大山芳史, 林良彦, “日本語語彙大系”, 岩波書店, (1997)

[3] Richard Zens, Franz Josef Och, Hermann Ney “Phrase-based Statistical Machine Translation”, KI 2002, pp35-56, (2002)

[4] giza-pp-v1.0.3.tar.gz http://www.fjoch.com/GIZA++.html

[5] Chiang,David. “A hierarchical phrase-based model for statistical machine translation.” In Proceedings of the 41nd Annual Meeting of the Association for Computational Linguistics (ACL05) .pp.263-270. (2005)

[6] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, “The Mathematics of Statistical Machine Translation: Parameter Estimation”, Association for Computational Linguistics 1993, pp263-311. (1993)

[7] Koehn, Philipp, H. Hoang, et al. Moses: “Open source toolkit for Statistical Machine Translation.” Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions， Prague， Czech Republic， pp. 177—180. (2007)

[8] joshua http://cs.jhu.edu/ ccb/joshua/

[9] Franz Josef Och “Minimum Error Rate Training in Statistical Machine Translation” Association for Computational Linguistics，pp.160-167. (2003)

[10] Chiang,David.“Hierarchical Phrase-Based Translation，” Computational Linguistics，33(2). 2007

[11] 西山七絵, 村上仁一, 徳久雅人, 池原悟, “単文文型パターン辞書の構築”, 言語処理学会第 11 回年次大会, pp.372-375. (2005)

[12] 村上仁一, 池原悟, 徳久雅人, “日本語英語の文対応の対訳データベースの作成”, 「言語，認識，表現」第 7 回年次研究会 (2002)

[13] Kishore Papineni etc.“BLEU: a Method for Automatic Evaluation of Machine Translation”, Association for Computational Linguistics, pp.311-318, (2002)

[14] “Automatic Evaluation of Machine Translation Quality Using *N*-gram CoOccurrence Statistics．” http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf. (2002)

[15] METEOR “The METEOR Automatic Machine Translation Evaluation System” http://www.cs.cmu.edu/ alavie/

[16] INPACT “Automatic Evaluation for Machine Translation” http://www.eli.hokkai-s-u.ac.jp/ echi

[17] RIBES “Rank-based Intuitive Bilingual Evaluation Measure” http://www.kecl.ntt.co.jp/icl/lirg/ribes

[18] Gregor Leusch， Nicola Ueffing and Hermann Ney. “A Novel String-to-String Distance Measure with Applications to Machine Translation Evaluation.” In Proc. of MT Summit IX， 240—247. TRANSLATION ERROR RATE (TER) 7.0 http://www.cs.umd.edu/ snover/tercom/ (2003)

[19] MeCab http://mecab.sourceforge.net/

[20] Tsuyoshi Okita, Andy Way “Statistical Machine Translation with Terminology”

[21] 乗松 潤矢 “統計的機械翻訳における階層フレーズモデルの書換え規則の検討”

[22] SRILM “The SRI Language Model Toolkit” http://www.speech.sri.com/projects/srilm