概要

近年，機械翻訳において，統計翻訳が注目されている．統計翻訳では，対訳コーパスから自動的に翻訳規則を獲得し，翻訳を行う．そのため，統計翻訳における翻訳品質は，対訳コーパスの量に大きく依存する．しかし，対訳コーパスの作成には，モノリンガルコーパスに比べて膨大なコストがかかる．

そこで本研究では，対訳コーパスと比較して作成が容易であるモノリンガルコーパスを用いる手法を提案する．はじめに，統計翻訳を用いて大量のモノリンガルコーパスを翻訳する．そして，翻訳文から精度の高い文を抽出し，対訳コーパスに加えることで，翻訳精度の向上を目指した．実験の結果，既存の対訳コーパスを用いた統計翻訳と比較して，BLEU，NIST，METEOR の値において，わずかに向上が認められた．向上がわずかであった原因として，モノリンガルコーパスを翻訳する際の翻訳精度の低さが考えられる．また，翻訳文からの抽出において，精度の高い文の抽出が不十分であることが考えられる．

よって今後は，より翻訳精度の高い文の抽出方法を検討する．また，より大量のモノリンガルコーパスを用いた実験を行うことを考えている．

目 次

図 目 次

表 目 次

1　はじめに

機械翻訳において，人手で翻訳規則を定義し，翻訳を行うルールベース翻訳が一般的であった．しかし，人手で翻訳規則を定義するには，莫大なコストがかかる．また，言語毎に文法規則が異なるため，多言語への拡張が困難であった．そこで近年では，統計翻訳が主流となっている．統計翻訳では，対訳コーパスから，自動的に翻訳規則を獲得するため，ルールベース翻訳に比べコストが低い．また，多言語への拡張が容易である．ここで，対訳コーパスとは，二言語間における対訳データを 1 文対応でまとめたコーパスである．これまでの研究で，統計翻訳における翻訳品質は，対訳コーパスの量に大きく依存することが分かっている [1]．統計翻訳において，この対訳コーパスを如何に獲得するかが大きな課題となっている．日英翻訳においても，日英対訳コーパスの量は，欧米諸国の対訳コーパスの量と比較すると非常に少量であるため，さらなる日英対訳コーパスの獲得が望まれる．しかし，対訳コーパスの作成には，モノリンガルコーパスに比べて膨大なコストがかかるという問題がある．

この問題を解決するために，様々な研究がなされている．Xiaoguang らは，中英翻訳において，モノリンガルコーパスを，ルールベース翻訳を用いて翻訳し，モノリンガルコーパスとその翻訳文を対訳コーパスに加えることで翻訳精度の向上を試みた [2]．また，Holger は，仏英翻訳において，大量のモノリンガルコーパスを，統計翻訳を用いて翻訳することで，対訳コーパスを増加させた [3]．しかし，いずれも翻訳精度の向上はほとんど認められなかった．これは，モノリンガルコーパスの翻訳文全てを用いたためであると考えられる．

そこで本研究では，モノリンガルコーパスの翻訳文から精度の高い文を抽出し，対訳コーパスに加える手法を提案する．モノリンガルコーパスと，精度の高い翻訳文の対を学習データに加えることで，翻訳精度の向上を目指す．また，対訳辞書データを補うため，“英辞郎”[4] を用いる．翻訳対の量が多い英辞郎のデータを対訳コーパスに付与することで，統計翻訳の精度を向上させる．

本論文の構成を以下に示す．第 2 章で従来の日英統計翻訳システムについて説明し，第 3 章で提案手法のシステムについて説明する．そして，第 4 章では実験環境を，第 5 章で実験結果を示し，第 6 章で本研究の考察を述べ，第 7 章で今後の課題を述べる．

2　日英統計翻訳システム

2.1　概要

統計翻訳において,「単語に基づく統計翻訳」と,「句に基づく統計翻訳」がある．初期の統計翻訳は，単語に基づく統計翻訳であった．しかし，近年提案された句に基づく統計翻訳 [5] は，語順の並び替えや文脈における訳語の選択や翻訳精度において，単語に基づく統計翻訳よりも優れている．よって，現在は句に基づく統計翻訳が主流となっている．そのため，本研究で扱う統計翻訳システムにおいても，句に基づく統計翻訳を用いる．また統計翻訳の特徴として，文法構造が似ている言語間では翻訳精度が高い傾向があり，文法構造の異なる言語間では翻訳精度が低い傾向がある．日英統計翻訳の枠組みを図 1 に示す． 日英統計翻訳では，日本語文 j を入力文とした場合に，翻訳モデル

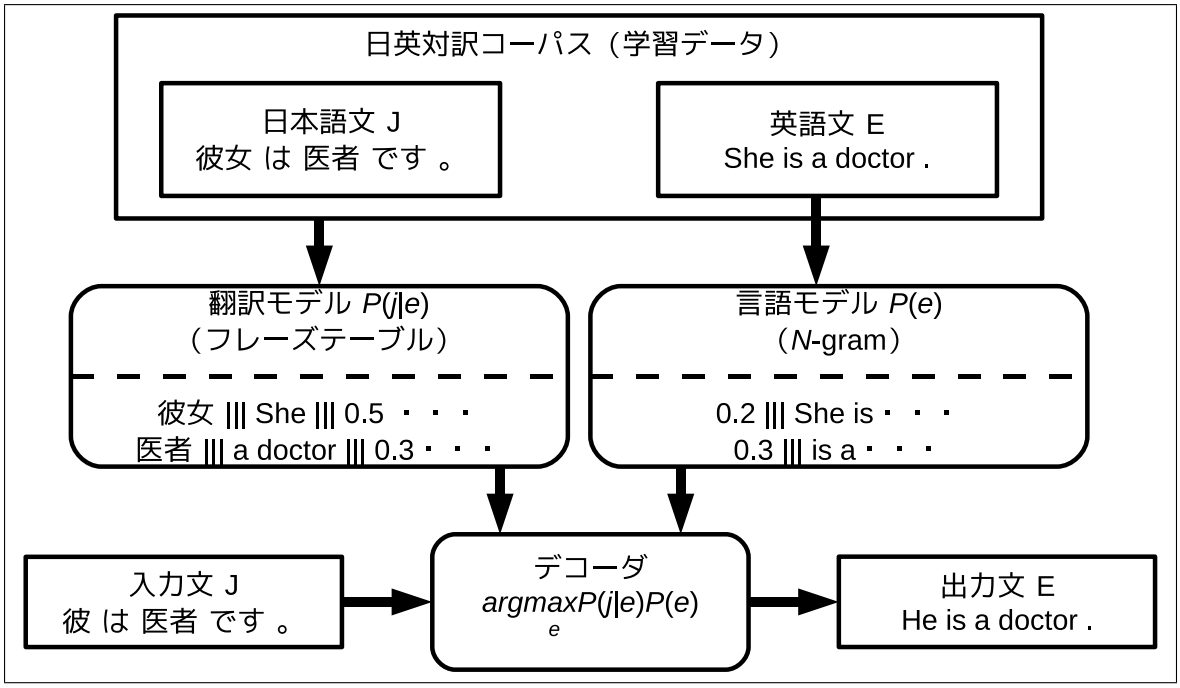

図 1: 日英統計翻訳の枠組

$P(j|e)$ と言語モデル $P(e)$ の全ての組み合わせから，確率が最大となる英語文 $\hat{e}$ を出力文とする．E を探索するシステムをデコーダと呼ぶ．以下に基本的なモデルを示す．

$$E = arg\max_{e} P(e|j) \tag{1}$$

$$\simeq arg\max_{e} P(j|e)P(e) \tag{2}$$

2.2 翻訳モデル

翻訳モデルは，日本語から英語の単語列へ，確率的に翻訳を行うためのモデルである．統計翻訳において，句に基づく翻訳モデルとして，一般的に，フレーズテーブルが用いられている．フレーズテーブルは以下の手順によって作成される．

手順 1 後述する IBM モデルを用いて，単語の対応を得る

手順 2 ヒューリスティックスなルールを用いて句に基づく対応を得る

手順 3 手順 2 で求めた句対応から，フレーズテーブルを作成する

詳しい作成手順については，後述する．また，表 1 にフレーズテーブルの例を示す．

表 1: フレーズテーブルの例

新緑 の 季節 ||| during the season of new green leaves ||| (0) (1) () ||| (0) (1) () () () () () ||| 0.333333 3.6554e-09 0.166667 1.83964e-19 2.718
新緑 の 季節 ||| during the season ||| (0) (1) () ||| (0) (1) () ||| 0.333333 3.6554e-09 0.166667 2.63823e-07 2.718
新緑 の 季節 に ||| during the season of new green leaves ||| (0) (1) () () ||| (0) (1) () () () () () ||| 0.333333 1.26847e-10 0.166667 1.83964e-19 2.718

左から順に，日本語フレーズ，英語フレーズ，フレーズ内単語対応（日英方向），フレーズ内単語対応（英日方向）日英方向の翻訳確率 $P(j|e)$，日英方向の単語の翻訳確率の積，英日方向の翻訳確率 $P(e|j)$，英日方向の単語の翻訳確率の積，フレーズペナルティである．ただし，フレーズペナルティの値は，常に自然対数の底である 2.718 である．

2.3 IBM 翻訳モデル

統計翻訳における単語対応を得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 [6] がある．この翻訳モデルは，提案者の Brown らが全員 IBM 社員であったため，IBM 翻訳モデルと呼ばれている．IBM 翻訳モデルは，$\mathrm{P}(F|E)$ の近似方法の違いによって，model1 から model5 までの，順に複雑になる 5 つのモデルから構成されている．各モデルのおおまかな違いを以下に示す．

model1 目的言語における，ある単語が原言語の単語に対応する確率のみを用いる

model2 model1 に加えて，目的言語における，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

model3 model2 に加えて，目的言語における，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 における permutation 確率を改良して用いる（model2 の絶対位置に対して，相対位置）

model5 model4 における permutation 確率を更に改良して用いる

IBM 翻訳モデルは，仏英翻訳を前提としている．しかし，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 モデルの基本的な計算式を示す．

$$P(J|E) = \sum_{a} P(J, a|E) \tag{3}$$

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．

$$P(j, a|E) = P(m|E) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, j_1^{j-1}, m, E) P(j_j|a_1^{j}, j_1^{j-1}, m, E) \tag{4}$$

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 ϵ 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．

$$P(J, a|E) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(j_j|e_{a_j}) \tag{5}$$

$$P(J|E) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(j_j|e_{a_j}) \tag{6}$$

$$= \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(j_j|e_i) \tag{7}$$

model1 において，翻訳確率 $t(j|e)$ の初期値が 0 でない場合には，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．

$$c(j|e; J, E) = \frac{t(j|e)}{t(j|e_0) + \cdots + t(j|e_l)} \sum_{j=1}^{m} \delta(j, j_j) \sum_{i=0}^{l} \delta(e, e_i) \tag{8}$$

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 λ_e を以下の式で計算する

$$\lambda_e = \sum_j \sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)}) \tag{9}$$

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

$$t(j|e) = \lambda_e^{-1} \sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)}) \tag{10}$$

$$= \frac{\sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)})}{\sum_j \sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)})} \tag{11}$$

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．

$$a(a_j|j,m,l) \equiv P(a_j|a_1^{j-1}, j_1^{j-1}, m, l) \tag{12}$$

よって，model1 の式 (6) は以下のように置き換えられる．

$$\begin{aligned} P(J|E) &= \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(j_j|e_{a_j}) a(a_j|j,m,l) && (13) \\ &= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(j_j|e_i) a(i|j,m,l) && (14) \end{aligned}$$

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．

$$\begin{aligned} c(j|e; J^{(s)}, E^{(s)}) &= \sum_{j=1}^{m} \sum_{i=0}^{l} \frac{t(j|e)a(i|j,m,l)\delta(j,j_j)\delta(e,e_i)}{t(j|e_0)a(0|j,m,l) + \cdots + t(j|e_l)a(l|j,m,l)} && (15) \\ c(i|j,m,l; J^{(s)}, E^{(s)}) &= \frac{t(j_j|e_i)a(i|j,m,l)}{t(j_j|e_0)a(0|j,m,l) + \cdots + t(j_j|e_l)a(l|j,m,l)} && (16) \end{aligned}$$

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，モデル 3 では単語の位置を絶対位置として考えている．モデル 3 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 ϕ 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 ϕ_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．

$$P(\phi_0|\phi_1^l, e) = \left(\begin{array}{c} \phi_1 + \cdots + \phi_l \\ \phi_0 \end{array} \right) p_0^{\phi_1 + \cdots + \phi_l - \phi_0} p_1^{\phi_0} \tag{17}$$

したがって，model3 は以下の式によって表される．

$$\begin{aligned} P(j|e) &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} P(j,a|e) && (18) \\ &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \left(\begin{array}{c} m - \phi_0 \\ \phi_0 \end{array} \right) p_0^{m - 2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i! n(\phi_i|e_i) \times \\ &\qquad \prod_{j=1}^{m} t(j_j|e_{a_j}) d(j|a_j, m, l) && (19) \end{aligned}$$

モデル 3 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

$$P(\Pi_{[i]1} = j|\pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_1(j - \odot_{i-1}|\mathcal{A}(e_{[i-1]}), \mathcal{B}(j_j)) \tag{20}$$

- それ以外の場合

$$P(\Pi_{[i]k} = j|{\pi_{[i]1}}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_{>1}(j - \pi_{[i]k-1}|\mathcal{B}(j_j)) \tag{21}$$

2.3.5 model5

モデル 4 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．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に制約が施されている．

2.4 GIZA++

GIZA++[7] とは，統計翻訳に用いるための単語の確率値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 ($p0$)，繁殖数が 1 である確率 ($p1$)，…，繁殖数が n である確率 (pn) で構成される．

2.5　フレーズテーブル作成法

IBM モデルは，方向のある 1 対多の単語アライメントである．よって，句レベルであるフレーズテーブルを得るには，両方向の 1 対多のアライメントを求める必要がある．

まず，GIZA++を用いて，学習文から日英，英日方向の最尤な単語アライメントを得る．日本語文 “風でろうそくが消えた” と，その対訳英語文 “The wind blew out the candle” を例に挙げ，図 2 と図 4 に日英方向の単語対応を示す．また，図 3 と図 5 に英日方向の単語対応を示す．なお，図 4 と図 5 において，●は対応点を示す．

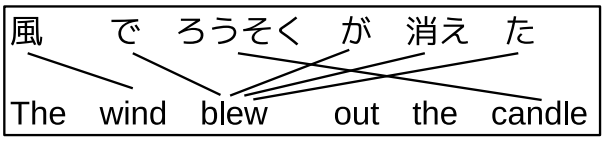

図 2: 日英方向の単語対応

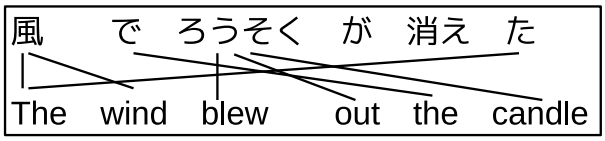

図 3: 英日方向の単語対応

	The	wind	blew	out	the	candle
風		●				
で			●			
ろうそく						●
が			●			
消え			●			
た			●			

図 4: 日英方向の単語対応

	The	wind	blew	out	the	candle
風	●	●				
で					●	
ろうそく			●	●		●
が						
消え						
た	●					

図 5: 英日方向の単語対応

次に，両方向のアライメントから，両方向に 1 対多の対応を認めた単語アライメントをヒューリスティックスなルールにより計算する．ここで，ヒューリスティックスとは，人間の日々の意思決定に類似した直感的かつ発見的な思考方法である．基本のヒューリスティックスとして，“intersection(積)”と，“union（和）”，“grow(成長)”，そして “grow-diag”がある．intersection は，両方向共に存在する対応点のみを用いる．また，union は，両方向の対応点を全て用いる．intersection の例を図 6 に，union の例を図 7 に示す．

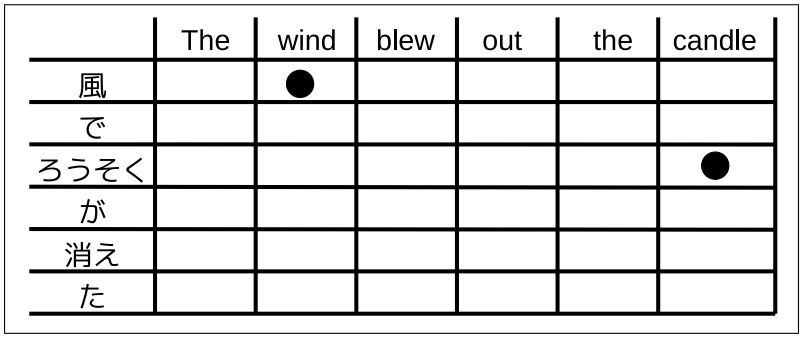

図 6: intersection の例

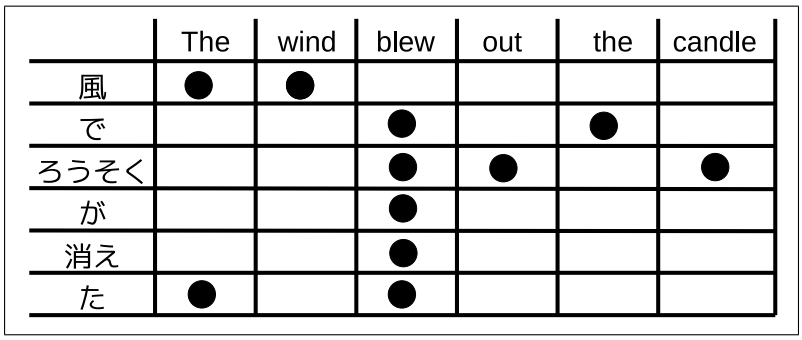

図 7: union の例

そして，grow, grow-diag は intersection と union の中間である．intersection からスタートし，既に採用した対応点の周りに union の対応点を加えていく．grow では縦と横の方向に，grow-diag では縦と横と対角に union の対応点がある場合に，その対応点を用いる．図 8 に grow の例を，図 9 に grow-diag の例を示す．なお，図 8 と 9 において，○は，intersection から追加された対応点を示す．

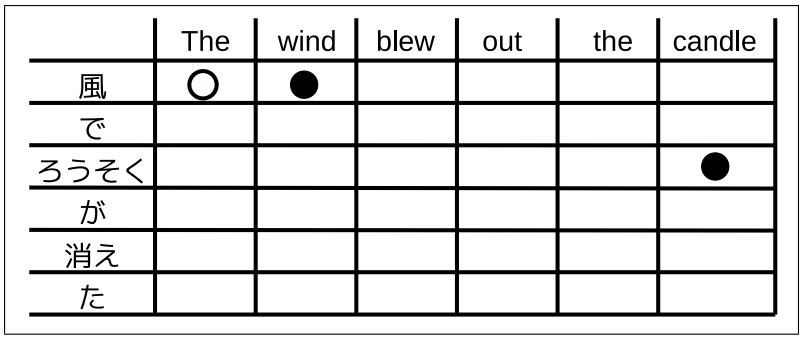

図 8: grow の例

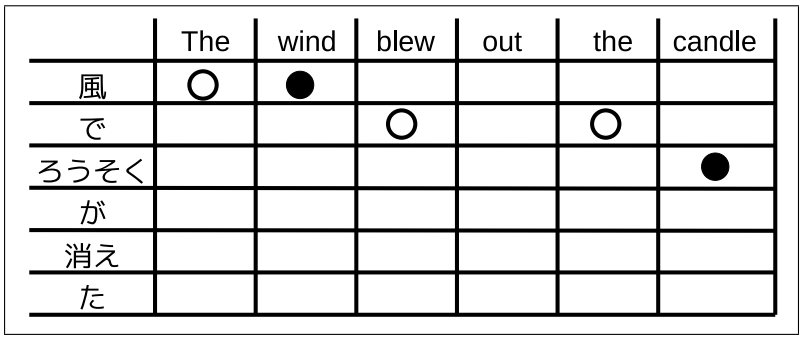

図 9: grow-diag の例

最後に，最終処理のヒューリスティックスとして，“final”と，“final-and”を用いる．finalは，少なくとも片方の言語の単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，両側言語の単語の単語対応がない場合に，unionの候補対応点を追加する．図10にgrow-diag-finalの例を，図11にgrow-diag-final-andの例を示す．ここでも，図10と11において，○は，grow-diagから追加された対応点を示す．

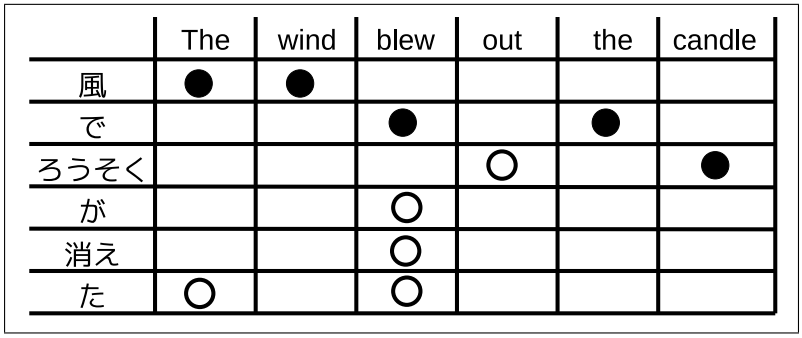

図 10: grow-diag-finalの例

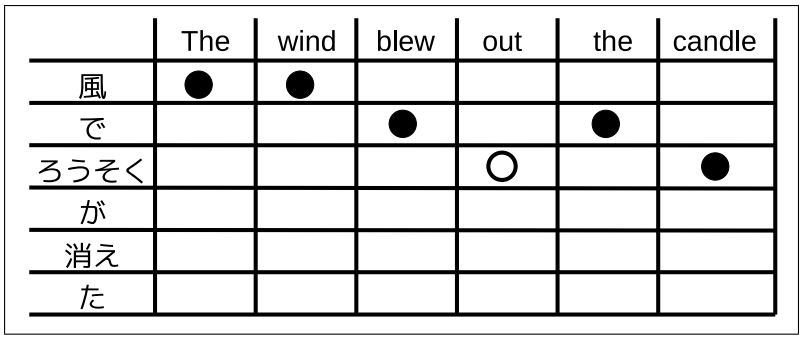

図 11: grow-diag-final-andの例

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．

2.6 言語モデル

言語モデルは，単語列の生成確率を付与するモデルである．日英翻訳では，翻訳モデルを用いて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ために用いる．統計翻訳では一般的に，N-gram モデルを用いる．表 2 に N-gram モデルの例を示す．

表 2: N-gram の例

-4.191673 the socket -0.3293359
-3.661356 the sofa -0.2541532
-3.70543 the software -0.08667657
-3.343311 the soil -0.3595161
-4.37106 the solar -0.09943552

N-gram モデルは，“単語列 $w_1^n = w_1, w_2, ..., w_n$ の i 番目の単語 w_i の生起確率 $P(w_i)$ は直前の $(n-1)$ 単語に依存する”，という仮説に基づくモデルである． 計算式を以下に示す．

$$P(w_1^n) = \Pi_n^{i=1} P(w_i \mid w_{i-1}) \tag{22}$$

例えば,「He is a doctor .」という文字列に対する 2-gram モデルは以下のようになる．

$$P(e = \text{“}He\, is\, a\, doctor\, .\text{”}) \approx P(He) \times P(is \mid He) \times P(a \mid is) \times P(doctor \mid a) \times P(. \mid doctor) \tag{23}$$

3-gram の場合を考えると，“He is” という単語列の次に “a” が来る確率を考える．しかし，N-gram モデルは局所的な情報であり，文法構造の情報を持たない．したがって，異なる文法構造間の翻訳は，同じ文法構造間の翻訳と比較して，翻訳精度が低下する傾向がある．

N-gram モデルにおいて，信頼できる値を算出するためには, 大規模なコーパスを用いることが必要である．そこで，出現数の少ない単語列をモデルの学習から削除する手法や，確率が 0 となるのを防ぐためのスムージング手法が提案されている．スムージングの代表的な手法としてバックオフスムージング (back-off smoothing) が挙げられる．バッ

クオフスムージングは学習データに出現しない N-gram の値をより低次の N-gram の値から推定する．trigram の場合の例を以下に示す．

$$P(w_i \mid w_{i-2}^{i-1}) = \begin{cases} \alpha \times p(w_i \mid w_{i-2}^{i-1}) & trigram \text{ が存在する場合} \\ \beta \times p(w_n|w_{n-1}) & trigram \text{ が存在せず, } bigram \text{ が存在する場合} \\ p(w_n|w_{n-1}) & \text{それ以外の場合} \end{cases} \quad (24)$$

ここで，α をディスカウント係数，β をバックオフ係数と呼ぶ．

2.7 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み合わせから，確率が最大となる翻訳候補を探索し，出力する．入力文として,「彼は医者です。」が入力されたときの翻訳例を図 12 に示す．

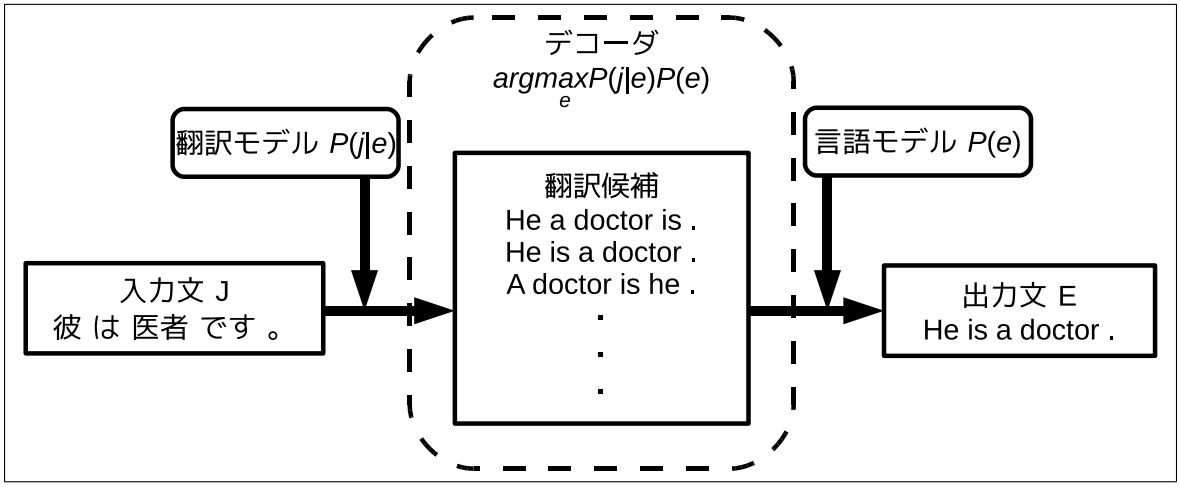

図 12: デコーダの動作例

デコーダは，日英統計翻訳において，$arg\max_e P(j|e)P(e)$ 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選択する必要がある．しかし，適切な英語文を決定するためには，莫大な計算量が必要となる．そこで莫大な計算量を削減するための手法として，ビームサーチ法やマルチスタック法が存在する．

2.8 パラメータチューニング

デコーダは, 言語モデルや翻訳モデルに対して重みを与えることができる．例えば，言語モデルに対して高い重みを与えると，デコーダは言語モデルの確率 $P(e)$ を重視した出力を行う．各モデルに与える重みをパラメータと呼ぶ．このパラメータを最適化するため，MERT(Minimum Error Rate Training)[8] という手法を用いる．MERT は，後述する自動評価法 BLEU のスコアが最大となる翻訳結果を出力するようにパラメータ $\hat{\lambda_1^n}$ の調整を行う．n 個のパラメータ $\hat{\lambda_1^n}$ の最適化に用いる式を以下に示す．

$$\hat{\lambda_1^n} = arg \max_{\lambda_1^n} BLEU(smt(\lambda_1^n), e_{ref}) \tag{25}$$

ここで， $smt(\lambda)$ はパラメータ λ が与えられたときの，デコーダの出力文である．また，$BLEU()$ は BLEU のスコアであり，デコーダの出力文と，入力文に対してあらかじめ用意された正解文 e_{ref} から計算される．なお，パラメータチューニングにおける入力文として，ディベロップメント文と呼ばれるデータを用いる．ディベロップメント文を試し翻訳し，各文に対して上位 N 個の翻訳候補を出力する．そして N 個の中から，より自動評価値が高い翻訳候補が上位に来るようにパラメータに $\hat{\lambda_1^n}$ 最適化する．試し翻訳とパラメータの調整を繰り返すことで，パラメータチューニングを行う．

3　提案手法

3.1　提案手法の概要

日英対訳コーパスは，日本語と英語の対訳文のコーパスである．また，日本語学習文と，英語学習文はそれぞれの言語のモノリンガルコーパスである．本研究では，対訳コーパスを増加させるため，はじめに，統計翻訳を用いて日本語学習文を翻訳する．次に，プログラムを用いて，翻訳文から精度の高い文を抽出する．そして，日本語学習文と抽出した文との対を対訳コーパスに加える．

3.2　提案手法の手順

提案手法の手順を図 13 に示す．

準備 大量のモノリンガルコーパスとして，日本語学習文と英語学習文を準備する．また，統計翻訳に用いる日英対訳コーパスと英辞郎のデータを準備する．

手順 1 日英対訳コーパスと英辞郎から，フレーズテーブルを作成する．また，英語学習文から N-gram を作成する．

手順 2 作成したフレーズテーブルと N-gram モデルを用いて，日本語学習文を翻訳する．

手順 3 翻訳文から尤度の高い文を抽出し，日本語学習文と対訳文とする．これを “抽出文対” と呼ぶ．

手順 4 抽出文対を日英対訳コーパスに付与し，新たなフレーズテーブルを作成する．

手順 5 手順 4 で作成したフレーズテーブルと，手順 1 で作成した N-gram を用いて日本語テスト文を翻訳する．

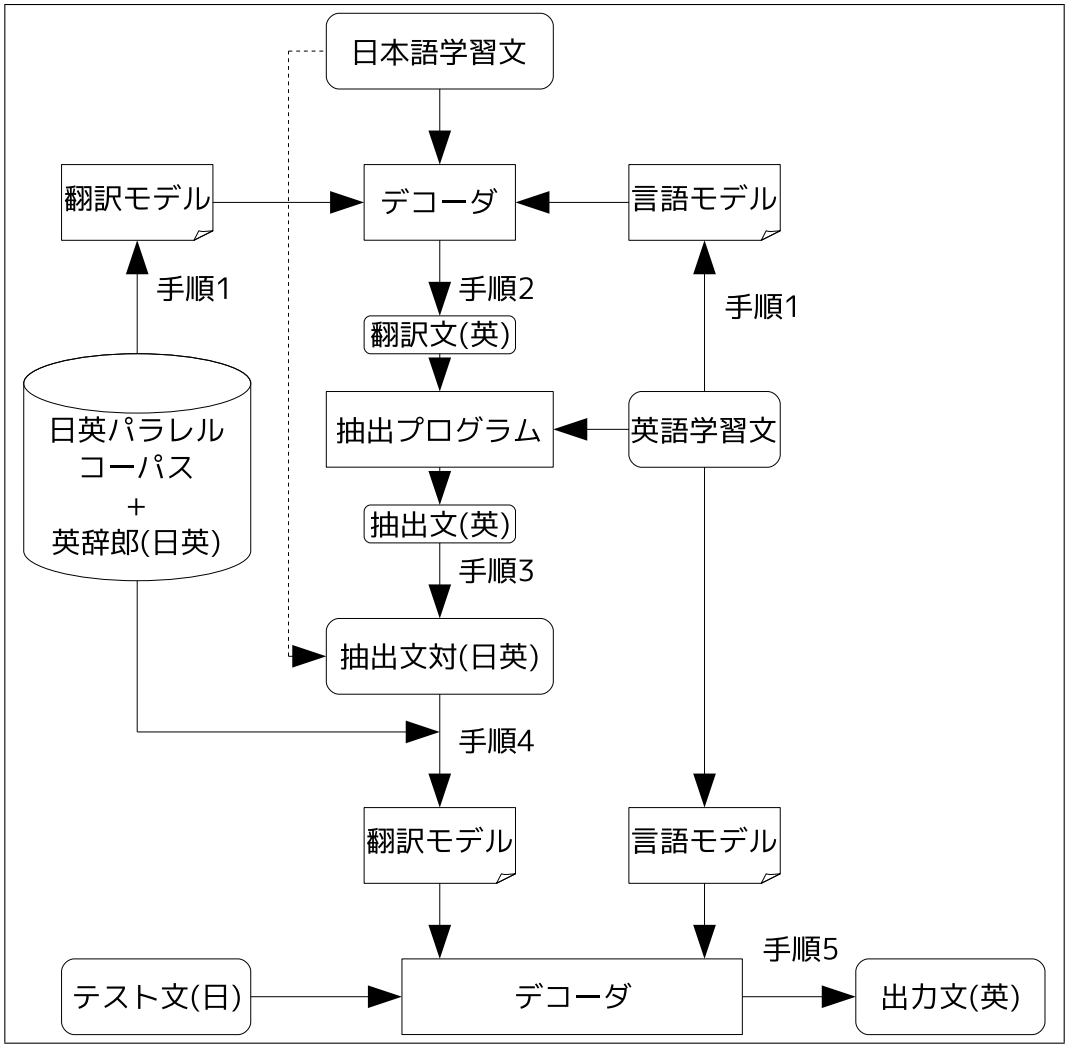

図 13: 提案手法の枠組

3.3 抽出方法

精度の高い文の抽出には，英語学習文から得た N-gram モデルを用いる．日本語学習文の翻訳文において，N-gram の尤度の高い文を抽出する．また，文の長さによる偏りを防ぐため，単語数で正規化を行う．抽出の際の閾値は，正しい文のまとまりである英語学習文の尤度の平均値とする．

4　実験環境

4.1　翻訳モデル

翻訳モデルの学習には，“GIZA++[7]” 用いる “train-factored-phrase-model.perl[9]” を用いる．なお，本研究では，ヒューリスティックスとして，“grow-diag-final-and” を用いる．

4.2　言語モデル

言語モデルの学習には，“SRILM[10]” の “ngram-count” を用いる．本研究では，N-gram モデルに 5-gram を用いる．

4.3　デコーダのパラメータ

デコーダには，“moses[9]” を用いる．また，moses の各パラメータは “mert-moses.pl[9]” を用いて最適化する．しかし，“ttable-limit” と “distortion-limit” についてはパラメータチューニングでは変更されない．“ttable-limit” とは，1 つの日本語のフレーズに対して考慮する，目的言語のフレーズ数の制限である．また，“distortion-limit” とは，フレーズの並び替えの範囲の制限である．本研究では，“ttable-limit” の値を 60，また “distortion-limit” の値を-1（無制限）とする．

4.4　実験データ

4.4.1　英辞郎

“英辞郎” は EDP(Electronic Dictionary Project) がアップデートし続けている英和・和英データベースである．英辞郎のデータには，通常の英語辞書にない新しい語彙や複雑な言い回しも含まれている．英辞郎のデータを学習データに加えることで対訳辞書データを補完する．本研究では，不適切なデータを除去し，学習データとして用いるためにクリーニングした 1,366,575 文対 [11] を用いる．表 3 にクリーニング前のデータ例を，表 4 に，クリーニング後の英辞郎のデータ例を示す．

表 3: クリーニング前の英辞郎データ例

■あなた
・bubeleh《イディッシュ》
・darling〔夫婦間や恋人同士の呼びかけ〕
・gentle reader〔作家が著作の中で読者に語りかける場合の「あなた」〕
■理解する
catch on（～の意味を）
put the pieces together（断片的な情報などを総合して）

表 4: クリーニング後の英辞郎データ例

あなた ||| bubeleh
あなた ||| darling
あなた ||| gentle reader
理解する ||| catch on
理解する ||| put the pieces together

4.4.2 単文コーパス

本研究では，実験に単文のみを用いる．単文の本来の意味は，主語と述語の関係が 1 回のみ成り立つ文である．しかし，本研究で用いる単文は，形態素解析器を用いて形態素解析した際に動詞が 1 つの文を抽出したものである．例えば,「彼は生き返った。」という文は，本来ならば単文であるが，形態素解析において,「彼/は/生き/返っ/た/。」と解析された場合には,「生き返る」という動詞ではなく,「生きる」と「返る」の 2 つの動詞が含まれているとみなして，本研究には用いない．以下に，本研究で用いる単文コーパスの例を示す．

表 5: 単文コーパスの例：日本語文

誰だって 1 人ではできない。
彼女は音楽の先生をしている。
それはできない相談だ。

本研究では，辞書の例文より抽出した単文コーパス 181,988 文 [12] から，以下のように用いる．

表 6: 単文コーパスの例：英語文

No one man can do it .
She is a music teacher .
That's an impossible proposition .

- 日英対訳コーパス：50,000 文対
- 英語学習文：100,000 文
- 日本語学習文：100,000 文
- テスト文：10,000 文
- ディベロップメント文：2,000 文（日本語学習文の翻訳に 1,000 文，テスト文の翻訳に 1,000 文）

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[13]”を用いて形態素解析を行う．また，英語文に対して “tokenizer.perl[9]”を用いて分かち書きを行う．

4.5 評価方法

4.5.1 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，用意された正解文と，機械翻訳システムが出力した出力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が，本研究では，BLEU(BiLingual Evaluation Understudy)[14]，NIST[15]，METEOR(Metric for Evaluation of Translation with Explicit ORdering)[16] を用いる．

- BLEU[14]
  BLEU は，機械翻訳の分野において，最も一般的な自動評価基準である．BLEU は，n-gram マッチ率に基づく手法を用いている．以下に式を示す．

$$BLEU = BPexp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{26}$$

$$w_n = 1/N \tag{27}$$

$$p_n = \frac{\sum_i \text{出力文中 } i \text{ と正解文中 } i \text{ で一致した } n-gram \text{ 数}}{\sum_i \text{出力文 } i \text{ 中の全 } n-gram \text{ 数}} \tag{28}$$

  ここで，式 (26) の BP は，機械翻訳の出力文が，正解文よりも短い場合のペナルティである．また，BLEU は文単位ではなくドキュメント単位での使用を前提としている．そのため，本研究では，文単位で使用するために変更が行われている．つまり，$P_n = 1$ の場合に BLEU 値が 0 になってしまうため，$P_n \neq 0$ であるような最大の n を N として使用する．BLEU は 0 から 1 の値を出力し，スコアが大きいほど評価が良い．なお，本研究では，BLEU 値の n-gram に 4-gram を用いている．

- NIST[15]
  NIST は，BLEU と同様に n-gram マッチ率を用いた手法である．しかし，式 (30) に示す情報量の計算式で得られる情報量によって重み付けしている点が異なる．また，ペナルティ関数も BLEU のペナルティ関数と異なっており，NIST の方が文長を考慮するようにペナルティがつけられる．NIST は，0 から∞の値を出力し，スコアが大きいほど評価が良い．

$$NIST = BP \sum_{n=1}^{N} \frac{\left(\sum_{\text{出力文 } i \text{ と正解文 } i \text{ に共通する } w_i \cdots w_n} Info_i\left(w_i \cdots w_n\right) \right)}{\sum_i \text{出力文 } i \text{ 中の全 } n-gram \text{ 数}} \tag{29}$$

$$Info(w_i \cdots w_i) = \log_2 \frac{\text{評価コーパス中の } w_1 \cdots w_{n-1} \text{数}}{\text{評価コーパス中の } w_1 \cdots w_n \text{数}} \tag{30}$$

- METEOR[16]
  METEOR は，再現率 R と適合率 P に基づく F 値に対して単語の非連続性に対するペナルティ関数 Pen を利用した評価基準である．式 32 のペナルティ関数 Pen にある m は機械翻訳の出力文と正解文との間で一致した単語数であり，c は一致した各単語を対象として語順が同じものを 1 つのまとまりとして統合した場合のまとまりの数である．したがって，機械翻訳の出力文と正解文が完全一致の場合には $c = 1$ となり，語順が全て逆の場合には c=m となる．α，β，γ の値はパラメータである．METEOR は，0 から 1 の値をスコアとして出力し，スコアが高いほど評価が良い．

$$F = \frac{P \times R}{\alpha \times P + (1-\alpha) \times R} \tag{31}$$

$$Pen = \gamma \times (c/m)^{\beta} \tag{32}$$

$$METEOR = (1 - Pen) \times F \tag{33}$$

4.5.2 人手評価

人手による評価として，対比較評価を行う．対比較評価では，“入力文”，“正解文”，“ベースライン出力文”，“提案手法出力文”が与えられ，ベースライン出力文と提案手法出力文の比較を行う．自動評価では，正解文に近い文の評価が高い．一方，人手評価では，入力文の翻訳文として確からしい文の評価を行う．しかし，人手評価には大きなコストがかかる．

5　実験結果

5.1　自動評価

テスト文 10,000 文を入力文として翻訳実験を行い，出力文に対して自動評価を行った．表 7 に，自動評価の結果を示す．

表 7 中の “ベースライン” とは，既存の対訳コーパスのみを用いて統計翻訳を行った結果である．また，“提案手法” とは，本研究で提案した手法の結果である．なお，提案手法において，既存の対訳コーパスに付与した “抽出文対” は 50,117 文対であった．

表 7: 自動評価結果

	BLEU	NIST	METEOR
ベースライン	0.1216	4.719	0.4990
提案手法	0.1241	4.720	0.4999

表 7 の結果より，提案手法が，ベースラインと比較してわずかに高い値を示していることが確認できる．

5.2 対比較評価

5.2.1 評価結果

ベースラインと提案手法の出力文から，それぞれランダムに抽出した 100 文を用いて，人手による対比較評価を行った．評価の基準を以下に示す．また，評価結果を表 8 に示す．

- 提案手法○　　：提案手法の方が良い
- 提案手法×　　：提案手法の方が悪い
- 差なし　　：双方とも意味が分からない，または，意味に差がない
- 同一出力　　：完全に同じ文が出力されている

表 8: 対比較評価

提案手法○	提案手法×	差なし	同一出力
8/100	12/100	50/100	30/100

結果より，人手評価において，提案手法の有効性は認められなかった．

5.2.2 翻訳例

提案手法○の翻訳例を表 9，表 10，表 11 に，提案手法×の翻訳例を表 12，表 13，表 14 に，差なしの場合の翻訳例を表 15，表 16，表 17 に示す．

- 提案手法○の翻訳例

  表 9 において，ベースラインに動詞がなく，提案手法には動詞があるため，提案手法○とした．

表 9: 提案手法○の翻訳例

入力文	警官 が 交通 整理 を し た 。
正解文	The police kept a clear passage for the traffic .
ベースライン	The policeman the traffic .
提案手法	A policeman is directing the traffic .

表 10 において，ベースラインに主語がなく，提案手法には主語があるため，提案手法○とした．

表 10: 提案手法○の翻訳例

入力文	関節 が 痛む 。
正解文	I ache in my joints .
ベースライン	joint aches .
提案手法	I have a pain in joint .

表 11 において，ベースラインの動詞 “is”に対して，提案手法の動詞句 “got out”が適切であると判断し，提案手法○とした．

表 11: 提案手法○の翻訳例

入力文	彼 は ビル から 外 に 出 た 。
正解文	He left the building to go outside .
ベースライン	He is out of the building .
提案手法	He got out of the building .

- 提案手法×の翻訳例

表 12 において，提案手法の翻訳文の意味が入力文と異なるため，不適切である．よって，提案手法×とした．

表 12: 提案手法×の翻訳例

入力文	物理 学 の 勉強 に は 数学 の 十分 な 知識 が 必要 で ある 。
正解文	The study of physics demands a good knowledge of mathematics .
ベースライン	The physics is necessary to the study of sufficient knowledge of mathematics .
提案手法	He is necessary to the study of physics sufficient knowledge of mathematics .

表 13 において，時制がベースラインの方が正しいため，提案手法×とした．

表 13: 提案手法×の翻訳例

入力文	気温 が ちょっと 上がっ た 。
正解文	The temperature rose a little .
ベースライン	The temperature went up a little .
提案手法	The temperature is going up a little .

表 14 において，提案手法の “stopping” と比較して，ベースラインの “staying” が適切であると判断し，提案手法×とした．

表 14: 提案手法×の翻訳例

入力文	ヒルトン ホテル に 泊まっ て い ます 。
正解文	I’m staying at the Hilton Hotel .
ベースライン	I am staying at Hilton .
提案手法	I am stopping with Hilton .

- 差なしの翻訳例

表 15 において，どちらの翻訳文も意味を成さないため，差なしとした．

表 15: 差なしの翻訳例

入力文	学会 で 研究 を 発表 する 。
正解文	Present one's research at the conference .
ベースライン	read at the meeting .
提案手法	I read .

表 16 において，“plenty of time” と “enough time” が入力文に対してどちらでも正しいといえるので，差なしとした．

表 16: 差なしの翻訳例

入力文	時間 は まだ 十分 ある 。
正解文	There is still plenty of time left .
ベースライン	There is still plenty of time .
提案手法	There is still enough time .

表 17 において，ベースラインの “six” と，提案手法の “six o'clock” は，どちらでも適切であると判断し，差なしとした．

表 17: 差なしの翻訳例

入力文	彼 は 約束 どおり 6 時 に 現れ た 。
正解文	He appeared at six as promised .
ベースライン	He showed up at six .
提案手法	He showed up at six o'clock .

6　考察

6.1　正しい対訳文を用いた場合の翻訳精度

日本語学習文と，正しい対訳文の対 100,000 文対を対訳コーパスに加えた場合を，“正しい対訳文対付与”とし，実験を行った．実験結果を表 18 に示す．

表 18: 正しい対訳文対付与

	BLEU	NIST	METEOR
ベースライン	0.1216	4.719	0.4990
正しい対訳文対付与	0.1562	5.283	0.5364

結果より，正しい対訳文対を学習データに加えると，評価値は大きく向上する．しかし，提案手法では翻訳精度の向上はほとんど認められなかった．この結果から，抽出文対の精度が不十分であると考えている．抽出文対には表 19 に示すような，誤りのある文が含まれている．誤りのある抽出文対を学習データとして用いた場合に，翻訳精度が下がると考えられる．したがって今後は，より精度の高い文の抽出方法を検討する必要がある．

表 19: 抽出文対の例

入力文	金 の 価値 が 上昇 し た 。
抽出文	The value of money .
入力文	こんな 品 が 手 に 入っ た 。
抽出文	I can’t work .

6.2　抽出の効果

抽出の効果を調査するため，システム全体の翻訳精度ではなく，モノリンガルコーパスの翻訳文の翻訳精度の調査を行った．日本語学習文の翻訳文 100,000 文の精度と，提案手法によって抽出された 50,117 文の精度を比較する．結果を表 20 に示す．

結果より，抽出により，精度の高いと思われる文を選出した方が評価値が高いという結果になった．したがって，抽出の有効性が確認できる．

表 20: 抽出の効果

抽出文数	BLEU	NIST	METEOR
100,000 文	0.1408	5.0951	0.5110
50,117 文	0.1635	5.3306	0.5522

6.3 抽出量の影響

誤りのある文の割合が，学習に及ぼす影響を調査するため，抽出の際の尤度を調整して実験を行った．表 21 に抽出文対数を 10,000 文対，20,000 文対，40,000 文対，80,000 文対，100,000 文対（全抽出）とした場合の自動評価の結果を示す．抽出文対数が多いほど誤りのある文の割合が高く，抽出文対数が少ないほど誤りのある文の割合が低い．ただし，本節の実験において，デコーダのパラメータによる評価結果のばらつきをなくすため，パラメータチューニングは行っていない．なお，“ベースライン”においても，抽出文対の付与を行わず，パラメータチューニングも行っていない．

表 21: 抽出量の影響

抽出文対数	BLEU	NIST	METEOR
ベースライン	0.0968	3.621	0.4407
10,000	0.0975	3.581	0.4405
20,000	0.0952	3.503	0.4362
40,000	0.0917	3.331	0.4276
80,000	0.0826	2.994	0.4122
100,000	0.0816	2.881	0.4062

結果より，学習データに誤りのある文がより多く含まれるほど，評価が下がることが確認できる．また，この結果からも，尤度を用いた抽出の有効性が示されたといえる．

6.4 モノリンガルコーパスの量

6.3 節において，尤度を高く設定し，抽出文対数を少なくすれば，誤りのある文の割合が減少し，評価結果が良くなることが示された．しかし，対訳コーパスに付与する抽出文対数が少ないと，学習に与える影響も小さい．したがって，より多くの日本語学習文の翻訳文から抽出を行えば，対訳コーパスに付与する際に，精度の高い対訳文が増加

し，提案手法の翻訳精度が向上すると考えられる．モノリンガルコーパスの収集は比較的容易に行うことができる．よって今後は，提案手法において，より大量のモノリンガルコーパスを用いた場合の，翻訳精度の調査を行う．

6.5 英辞郎の効果

提案手法において，学習データに英辞郎を用いた場合と，用いない場合における出力文中の未知語数を表 22 に示す．また，それぞれの場合における自動評価の結果を表 23 に示す．なお，対訳コーパスに付与した抽出文対はそれぞれ約 50,000 文対である．

表 22: 出力文中の未知語数

	未知語数
提案手法	1520
英辞郎なし	5587

表 23: 自動評価結果

	BLEU	NIST	METEOR
提案手法	0.1241	4.720	0.4999
英辞郎なし	0.1125	4.332	0.4730

結果より，学習データに英辞郎を加えることで，未知語が減少し，翻訳精度が向上している．したがって，本研究における英辞郎の有効性が確認できる．

6.6 ルールベース翻訳の併用

6.6.1 ルールベース翻訳

ルールベース翻訳とは，言語の専門家などによって厳密に定義された文法のルールを用いて翻訳を行う翻訳手法である．長所としては，ルールが存在する文の翻訳における精度が高いことが挙げられる．しかし短所として，ルールが存在しない場合には翻訳精度が低いことが挙げられる．また，人手で文法ルールを生成するため，コストや時間がかかることが挙げられる．本節では，ルールベース翻訳として，市販されている翻訳ソフト “翻訳の王様 Version5(IBM)” を用いる．なお，市販の翻訳器では独自の形態素解析を行う．そのため，市販の翻訳器を用いる際には形態素解析前の日本語文コーパスを使用する．

6.6.2 ルールベース併用

モノリンガルコーパスの翻訳を，ルールベースを用いて翻訳を行う実験 “ルールベース併用” を行った．本手法の統計翻訳システムと，全く別のシステムであるルールベール翻訳システムを用いることで，翻訳精度の向上するのではないかと考えた．なお，ルールベース併用において，抽出文対は 47,189 であった．表 24 に結果を示す．また，提案手法とルールベース併用それぞれの場合の未知語数を表 25 に示す．

表 24: ルールベース併用

	BLEU	NIST	METEOR
提案手法	0.1241	4.720	0.4999
ルールベース併用	0.1262	4.665	0.4950

表 25: 出力文中の未知語数

	未知語数
提案手法	1520
ルールベース併用	1472

表 24 の結果より，ルールベース併用において，提案手法との違いは認められなかった．また，表 25 より，未知語数の減少はわずかであった．

6.7 出力文の解析

ベースラインと，提案手法の出力文において，使用されたフレーズテーブルの解析を行った．

- 解析例 1

表 26: 解析文例 1

入力文	警官 が 交通 整理 を し た 。
正解文	The police kept a clear passage for the traffic .
ベースライン	The policeman the traffic .
提案手法	A policeman is directing the traffic .

表 26 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 27: 解析文例 1 のベースラインで使用されたフレーズテーブル

が ||| The
警官 ||| policeman
交通 整理 を し ||| the traffic
た 。 ||| .

表 28: 解析文例 1 の提案手法で使用されたフレーズテーブル

警官 が 交通 整理 を し ||| A policeman is directing the traffic
た 。 ||| .

解析文例 1 では，提案手法において，非常に長いフレーズが選出されている．原因として，抽出文対の中に，“警官”や，“交通 整理 を し”のフレーズが含まれていたが，対訳の英語文が異なっていたことが考えられる．“警官”が “policeman”に翻訳される確率が下がったことで，提案手法において長いフレーズが選出されたと考えられる．

- 解析例 2

表 29: 解析文例 2

入力文	関節 が 痛む 。
正解文	I ache in my joints .
ベースライン	joint aches .
提案手法	I have a pain in joint .

表 29 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 30: 解析文例 2 のベースラインで使用されたフレーズテーブル

関節 ||| joint
が 痛む 。 ||| aches .

表 31: 解析文例 2 の提案手法で使用されたフレーズテーブル

が 痛む ||| I have a pain in
関節 ||| joint
。 ||| .

提案手法において，“が 痛む”に対して，“I have a pain in”という長いフレーズが選出されている．この原因としても，解析文例 1 と同様に，“が 痛む”のいくつかの対訳フレーズの確率が，抽出文対によって変動したことが考えられる．

- 解析例 3

表 32: 解析文例 3

入力文	彼 は ビル から 外 に 出 た 。
正解文	He left the building to go outside .
ベースライン	He is out of the building .
提案手法	He got out of the building .

表 32 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 33: 解析文例 3 のベースラインで使用されたフレーズテーブル

彼 は ||| He is
から 外 に 出 ||| out of the
ビル ||| building
た 。 ||| .

表 34: 解析文例 3 の提案手法で使用されたフレーズテーブル

彼 は ||| He
から 外 に 出 ||| got out of the
ビル ||| building
た 。 ||| .

解析文例 3 では，ベースラインにおいて，“彼 は” が “He is ” に翻訳されたことで，不自然な翻訳文となった．しかし，提案手法において，“彼 は” が “He”に翻訳されている．これは，抽出文対中に，“彼 は”が “He”に翻訳されている例が多くあったことを示していると考えられる．

- 解析例 4

表 35: 解析文例 4

入力文	物理 学 の 勉強 に は 数学 の 十分 な 知識 が 必要 で ある 。
正解文	The study of physics demands a good knowledge of mathematics .
ベースライン	The physics is necessary to the study of sufficient knowledge of mathematics .
提案手法	He is necessary to the study of physics sufficient knowledge of mathematics .

表 35 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 36: 解析文例 4 のベースラインで使用されたフレーズテーブル

は ||| The
物理 学 ||| physics
が 必要 で ある ||| is necessary
の 勉強 に ||| to the study of
十分 な 知識 ||| sufficient knowledge
の ||| of
数学 ||| mathematics
。 ||| .

表 37: 解析文例 4 の提案手法で使用されたフレーズテーブル

は ||| He
が 必要 で ある ||| is necessary
の 勉強 に ||| to the study of
物理 学 ||| physics
十分 な 知識 ||| sufficient knowledge
数学 の ||| of mathematics
。 ||| .

解析文例 4 では，提案手法において，“は”が “He” に翻訳されたことで，適切ではない英語文が出力された．この原因の 1 つとして，抽出文対中の誤りのある文によって，“は”が “He” に翻訳される確率が高くなったことが考えられる．

- 解析例 5

表 38: 解析文例 5

入力文	気温 が ちょっと 上がっ た 。
正解文	The temperature rose a little .
ベースライン	The temperature went up a little .
提案手法	The temperature is going up a little .

表 38 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 39: 解析文例 5 のベースラインで使用されたフレーズテーブル

が	The
気温	temperature
上がっ た	went up
ちょっと	a little
。	.

表 40: 解析文例 5 の提案手法で使用されたフレーズテーブル

気温 が	The temperature is
上がっ	going up
ちょっと	a little
た 。	.

解析文例 5 では，ベースラインにおいて，“上がっ た” が “went up” と翻訳され，過去形になっている．しかし，提案手法では，“上がっ” が “going up” に翻訳され，進行形となっている．この例では，過去形が正しいと判断したが，文脈を考慮するなら進行形でも正しい文であるといえる．したがって，解析文例 5 において，ベースラインと提案手法では選出されたフレーズテーブルに大きな差はないといえる．

- 解析例 6

表 41: 解析文例 6

入力文	その 風習 は 5 0 年 ほど 前 に 滅び た 。
正解文	That custom died out about 50 years ago .
ベースライン	That practice has lost in 50 years ago .
提案手法	That practice has lost five hundred years ago .

表 41 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 42: 解析文例 6 のベースラインで使用されたフレーズテーブル

その 風習 は ||| That practice has
滅び ||| lost
に ||| in
5 0 年 ||| 50 years
ほど 前 ||| ago
た 。 ||| .

表 43: 解析文例 6 の提案手法で使用されたフレーズテーブル

その 風習 は ||| That practice has
滅び ||| lost
5 0 ||| five hundred
年 ほど 前 に ||| years ago
た 。 ||| .

提案手法において，数字の翻訳に誤りがある．この原因として，抽出文対中の誤りのある文の中に “5 0”に対して “five hundred”となる文が含まれていたことが考えられる．

- 解析例 7

表 44: 解析文例 7

入力文	涙 が 一 粒 彼女 の ほお を 流れ落ち た 。
正解文	After ran down her check .
ベースライン	Tears ran down her cheeks with a grain .
提案手法	Tears trickled down her cheeks . a grain .

表 44 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 45: 解析文例 7 のベースラインで使用されたフレーズテーブル

涙 が ||| Tears
流れ落ち ||| ran down
彼女 の ほお ||| her cheeks
一 ||| with
を ||| a
粒 ||| grain
た 。 ||| .

表 46: 解析文例 7 の提案手法で使用されたフレーズテーブル

涙 が ||| Tears
を 流れ落ち ||| trickled down
彼女 の ほお ||| her cheeks
。 ||| .
一 ||| a
粒 ||| grain
た ||| .

解析文例 7 において，ほとんどのフレーズは同じものか，同様の意味のものが用いられている．しかし，提案手法において，“た”が “. ”に翻訳されたため，不自然な翻訳文となった．この原因も，抽出文対中の誤りにあると思われる．

- 解析例 8

表 47: 解析文例 8

入力文	あの 人 は 決して 学者 など と は いえ ない 。
正解文	He has no claim to scholarship .
ベースライン	That person is never as a scholar .
提案手法	Time , however , is not do not that person .

表 47 のとき，ベースライン，提案手法で使用されたフレーズテーブルはそれぞれ以下の通りであった．

表 48: 解析文例 8 のベースラインで使用されたフレーズテーブル

あの 人 は ||| That person
は ||| is
決して ||| never
いえ ||| as
と ||| a
学者 など ||| scholar
ない 。 ||| .

表 49: 解析文例 8 の提案手法で使用されたフレーズテーブル

と は いえ ||| Time , however
など ||| ,
は ||| is
ない ||| not
決して 学者 ||| do not
あの 人 ||| that person
。 ||| .

提案手法において，“と は いえ”が “Time , however”に， “決して 学者” が “do not” に翻訳されたことによって不自然な英語文が出力されている．この原因も，抽出文の誤りによって，翻訳確率が変動したためであると考えられる．

- 解析における考察

  8件の解析を行ったが，いずれの例でも，抽出文対を学習データに用いることによって，使用されるフレーズテーブルが大きく変化していた．しかし，提案手法の出力が良い場合でも，あるフレーズの翻訳確率が分散することによって偶然に良いフレーズが選ばれるというような，抽出文対の誤りから発生する副次的なものであると考えられる．したがって，これまでの考察でも述べたように，抽出文対の精度を高めることが必要であると考えられる．

7　今後の課題

今後の課題として，以下を考えている．

- 抽出文対の精度向上
  提案手法において，抽出文対の精度を向上させることを考えている．現在検討している手法としては，以下の 2 つが挙げられる．まず，1 つめは，デコーダの翻訳確率を用いる手法である．N-gram モデルでは，単語の並びでのみ文を評価する．そこで，翻訳の際の翻訳確率を用いることで，翻訳文として確からしい文も選出できると考える．次に，構文解析器を用いる手法である．抽出文対には，用言が含まれていない文がみられる．よって，構文解析器によって用言がない文などを取り除けば，より精度の高い文の選出が可能であると考えている．

- 学習データ中の誤り割合による精度変化の調査
  学習データ中に含まれる誤りデータの割合で，どのように翻訳精度が変化するかを調査することを予定している．抽出文対中の誤りを完全に取り除くことは非常に困難である．そのため，学習データ中に，わずかでも誤りがあるだけで，翻訳精度が大きく減少するのであれば，抽出文対を加えて対訳コーパスを増加させることは困難であると考えられる．

- より大量のモノリンガルコーパスを用いた研究
  より大量の日本語学習文を用いて，本研究を行うことを考えている．モノリンガルコーパスの翻訳文から精度の高い文を抽出すると，対訳コーパスに付与できる文の数は減少する．よって，大量のモノリンガルコーパスを用いることで精度の高い対訳文対が大量に得られる可能性があると考えている．

8　おわりに

本研究では，モノリンガルコーパスの翻訳文から精度の高い文を抽出し，対訳コーパスに加えることで，対訳コーパスを増加させる手法を提案した．実験の結果，BLEU 値において 0.25%の向上，また，NIST，METEOR においてもわずかに翻訳精度の向上が認められた．わずかしか翻訳精度の向上が認められなかった原因として，抽出された文対のなかに，精度の低い文が含まれていたことが挙げられる．今後は，より精度の高い翻訳文の抽出方法を検討する．また，より大量のモノリンガルコーパスを用いて，翻訳精度を向上させる方法を検討する．

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計算機工学講座 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共に，厚く御礼申し上げます．そして，日常の議論を通じて多くの知識や示唆を頂いた同研究室の皆様に深謝いたします．また，参考にさせていただいた著書の著者の方々に，感謝の気持ちと御礼を申し上げたく，謝辞にかえさせていただきます．

参考文献

[1] 猪澤雅史, 村上仁一, 徳久雅人, 池原悟: “統計翻訳における単文・重文複文の翻訳精度の評価”，言語処理学会第 14 回年次大会, pp.869-872, 2008.

[2] Xiaoguang Hu, Haifeng Wang, Hua Wu: “Using RBMT Systems to Produce Bilingual Corpus for SMT”, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.287-195, 2007.

[3] Holger Schwenk: “Investigations on Large-Scale Lightly-Supervised Training for Statistical Machine Translation”, Proceedings of IWSLT 2008, 2008.

[4] 英辞郎 http://www.alc.co.jp/.

[5] Franz Josef Och, Hermann Ney: ”A Systematic Comparison of Various Statistical Alignment Models”, Computational Linguistics, volume 29, number 1, pp.19-51, March 2003.

[6] 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.

[7] GIZA++
http://www.fjoch.com/GIZA++

[8] Franz Josef Och: “Minimum Error Rate Training in Statistical Machine Translation”, In Proceeding of the 41st Annual Meeting of the Association for Computational Linguistics, pp.160-167, 2003.

[9] 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, pages 177-180, June 2007.

[10] SRILM(The SRI Language Modeling Toolkit) : srilm.tgz
http://www.speech.sri.com/projects/srilm/.

[11] 東江恵介, 村上仁一, 徳久雅人, 池原悟: “日英統計翻訳における英辞郎の効果”, 言語処理学会第 16 回年次大会発表論文集, pp.641-644, 2010.

[12] 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.

[13] Mecab：mecab-0.97.tar.gz，mecab-ipadic-2.7.0-20070801.tar.gz
http://mecab.sourceforge.net/.

[14] 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.

[15] NIST，Automatic Evalation of Machine Translation Quality Using n-gram Co-Occurence Statistics
http://www.itl.nist.gov/iad/mig/test/mt

[16] 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.