概要

近年，機械翻訳において，ニューラル機械翻訳 (Neural Machine Translation; NMT)[?] が注目されている．NMT は，対訳学習文とニューラルネットワークを用いて入力の原言語文に対して最尤となる出力の目的言語文を得る確率を学習する機械翻訳の手法である．本研究では，NMT の問題として，対訳学習文中に出現する頻度の低い単語 (以下，低頻度語) の問題に着目する．NMT におけるモデルの学習時，低頻度語を含む全語彙を学習した際に，システム全体の翻訳精度が低下するという可能性がある．この理由として，低頻度語は学習の確率値の統計的信頼性が低いこと，さらに対訳学習文中に多く出現することが考えられる．

そこで本研究では，NMT における新たな低頻度語処理の手法を提案し，システム全体の翻訳精度の向上を目指す．本研究の提案手法では，まず学習時に，語彙数制限処理として対訳学習文中に 1 回のみ出現する単語 (以下，頻度 1 単語) を全て特殊記号 (<unk>) に置換し，学習を行う．これは，頻度 1 単語が低頻度語の中でも特に統計的信頼性が低いと考えられるためである．次に翻訳時，Jean ら [?] による <unk> の置換処理を行い，出力文中の <unk> を Attention 確率が最も高い原言語単語 (以下，未知語) に置き換える．最後に未知語処理として出力文に含まれる未知語を対訳学習文と IBM Model 1 により作成した対訳単語辞書を用いて置換する [?] 手法を提案する．

結果として，出力文 100 文における人手評価結果では，低頻度語を含む全語彙を学習する方法と比較して，提案手法の方が良い例が 36 文，低頻度語を含む全語彙を学習する方法の方が良い例が 18 文となった．これより提案手法を用いることで，低頻度語を含む全語彙を学習する方法と比較して翻訳精度の向上が確認できた．

目 次

図 目 次

表 目 次

1　はじめに

機械翻訳の手法として，これまで，ルールベース翻訳，用例翻訳，統計翻訳などが提案されてきた．近年では，新たな機械翻訳の手法としてニューラル機械翻訳 (Neural Machine Translation; NMT) が注目されている．NMT の手法は，これまで提案された他の手法と比較して流暢性の高い翻訳文を生成することができると報告されている．

NMT において学習時の語彙数はニューラルネットワークの出力層の次元数に相当する．語彙数が多くなることは出力層が高次元になることに等しく，語彙数が多くなることで計算量が膨大となる．このため，計算量削減の目的で，対訳学習文中の一部の単語を特殊記号に置き換えることで，語彙数を制限する手法が用いられる．この際，対訳学習文中の頻度上位数万単語を利用し，それ以外の単語を特殊記号に置き換える方法が一般的である [?][?].

この手法の問題として，特殊記号は意味を持たない記号であるため，低頻度語を含む入力文に対する正しい出力が学習されないことが挙げられる．関沢ら [?] はこれに対して，対訳学習文中の低頻度語を特殊記号へ置換せず，同義の高頻度語に言い換える手法を提案している．しかし，この手法は対訳学習文の他に言い換え辞書を準備する必要がある．また，言い換えによる変換は，元の表現を完全に保持できるとは限らず，繰り返し言い換えを行うことで精度が低下する可能性がある．

本研究では，上記の問題に対して，低頻度語を含む全語彙を用いて学習を行う場合においても，翻訳精度が低下する可能性があることを指摘する．この理由として，低頻度語は学習の確率値の統計的信頼性が低いこと，さらに対訳学習文中の語彙の多くを占めることが挙げられる．以上を踏まえ，本研究では，NMT における低頻度語の問題を改善する新たな手法を提案する．提案手法により，低頻度語を含む全語彙を学習する方法と比較して翻訳精度の向上が確認できた．

本論文の構成を以下に示す．第 2 章でニューラル機械翻訳について，第 3 章で IBM 翻訳モデルについて，第 4 章で先行研究について説明する．そして，第 5 章で提案手法のシステムについて説明する．その後，第 6 章で実験環境を，第 7 章で実験結果を示し，第 8 章で本研究の考察を述べる．

2　ニューラル機械翻訳

2.1　概要

ニューラル機械翻訳 (NMT) とは，近年提案された機械翻訳の手法である．多くの細かい構成要素によって成立する従来の統計翻訳のシステムとは異なり，NMT は巨大なニューラルネットワークを一つ (もしくは少数) 用いてシステムを構築する．NMT の手法には，Encoder-Decoder モデルおよびそれを拡張した Attention モデルが提案されている．Encoder-Decoder モデルは入力の系列を固定長のベクトルに符号化 (Encode) し，固定長のベクトルより出力の系列を復号化 (decode) するモデルである．機械翻訳のタスクにおいて，Encoder-Decoder モデルの入出力系列の要素は単語のベクトル表現となる．Attention モデルは Encoder-Decoder モデルにおいて出力系列を生成する際に，入力系列の参照を行う機構 (Attention) を用いた手法であり，Encoder-Decoder モデルと比較して，長い入力に対するより妥当な出力を得られるとされている．

2.2 Encoder-Decoder モデル

Encoder-Decoder モデル [?] は 2 つのリカレントニューラルネットワーク (RNN) により構成されるニューラル機械翻訳のモデルである．1 つの RNN は入力系列を一つの固定長のベクトルに符号化 (encode) し，もう一方の RNN により固定長のベクトル符号を出力の系列へと復号化 (decode) する．Encoder-Decoder モデルでは，同時に 2 つの RNN の訓練を行い，入力の原言語文に対して出力の目的言語文を得る条件付き確率を最大化する．

2.2.1 Encoder-decoder モデルの枠組み

Encoder-Decoder モデルでは，Encoder は入力の系列 $\boldsymbol{x} = (x_1, \cdots, x_{Tx})$ をベクトル c に符号化する．一般的な Encoder-Decoder モデルでは**??**式および**??**式で表される RNN を用いて構成される．

$$h_t = f(x_t, h_{t-1}) \tag{1}$$

$$c = q(\{h_1, \cdots, h_{T_x}\}) \tag{2}$$

ここで h_t は各時刻 t の隠れ層の状態であり，c は隠れ層を用いて生成されたベクトルである．f および q は活性化関数であり，Sutskever ら [?] は f に LSTM を用いた上，$q = (\{h_1, \cdots, h_T\}) = h_T$ としている．Decoder は文脈ベクトル c と既に生成された単語 $\{y_1, \cdots, y_{t'-1}\}$ が与えられた際の次の単語 y_t を予測するように訓練され，結合確率を**??**式に示す条件式に分解することで翻訳文 $\boldsymbol{y} = (y_1, \cdots, y_{T_y})$ を得る条件付き確率を定義している．

$$p(\boldsymbol{y}) = \prod_{t=1}^{T} p(y_t | \{y_1, \cdots, y_{t-1}\}, c) \tag{3}$$

??式および**??**式で表される RNN を用いて，それぞれの条件付き確率は**??**式によりモデル化される．ここで，g は非線形の多層関数であり，y_t の確率を生成する．s_t は RNN の隠れ層となる．

$$p(y_t | \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \tag{4}$$

2.3 Attention モデル

Encoder-Decoder モデルの問題は，原言語文中の全ての情報を一つの固定長のベクトルに圧縮する点である．Encoder-Decoder モデルでは対訳学習文中で用いられている文よりも長い原言語文が入力された場合に，極端に精度が低下することが報告されている．これは，長い原言語文の全情報が一つの固定長のベクトルに圧縮されるために，目的言語文を生成する際に必要な情報が損失することが原因である．

Attention モデル [?] はこの問題を改善するために提案された NMT の手法である．Attention モデルの Encoder では入力単語を前後両方向から RNN に渡す手法 (bidirectional RNN) を用いている．Encoder は，入力文を前から読み込んだノードと後ろから読み込んだノードを組み合わせて各単語を符号化 (encode) する．Decoder では，モデルが出力文中のある単語を生成する際に，その単語が最も相関する原言語文中の単語に相当する符号を探索する．その後，探索により得られた原言語単語の文中の位置情報を有した文脈ベクトル，および既に生成された全ての目的言語単語を参照し，次の目的言語単語を予測する．Attention モデルは Encoder-Decoder モデルと異なり，原言語文の情報を一つの固定長のベクトルに圧縮せずに目的言語文中の単語生成時に参照することで，より長い文における精度の向上を実現している．

2.3.1 Attention モデルの枠組み

Attention モデルでは**??**式の条件付き確率を**??**式により定義する．

$$p(y_i|y_1, \cdots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, s_i, c_i) \tag{5}$$

ここで，s_i は時刻 i での隠れ層の状態であり，**??**式により計算される．

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{6}$$

確率 s_i は，Encoder-Decoder モデルと異なり，各目的言語単語 y_i について文脈ベクトル c_i により状態付けられる．文脈ベクトル c_i はアノテーション系列 $(h_i, \cdots, h_{T_x})$ に依存し，Encoder により入力文と対応付けられる．各アノテーション h_i は特に入力文中の i 番目の単語付近の情報を強く保有しており，さらに全入力文の情報を保持している．また，**??**式に示す文脈ベクトル c_i はアノテーション h_i の重み付き和により計算される．

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{7}$$

各アノテーション h_j の重み α_{ij} は

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{8}$$

により計算される．ここで

$$e_{ij} = a(s_{i-1}, h_j) \tag{9}$$

は j 番目付近の入力の対応および i 番目の出力の適合を示す値であり，アライメントモデルと呼ばれる．この値は y_i を出力する直前の RNN の隠れ層の状態 s_{i-1} および入力文中の j 番目のアノテーション h_j に基づいている．

3　IBM 翻訳モデル

以下，川原 [?] の論文を参照して記述している．統計翻訳における単語対応を獲得するための代表的なモデルとして，IBM の Brown らによる仏英翻訳モデル [?] がある．以下，IBM 翻訳モデルは，model1 から model5 までの 5 つのモデルから構成されている．各モデルの概要を以下に示す．

model1 目的言語のある単語が原言語の単語に訳される確率を用いる

model2 model1 に加えて，目的言語のある単語に対応する原言語の単語の原言語文中での位置の確率（以下，permutation 確率と呼ぶ）を用いる（絶対位置）

model3 model2 に加えて，目的言語のある単語が原言語の何単語に対応するかの確率を用いる

model4 model3 の permutation 確率を改良（相対位置）

model5 model4 の permutation 確率を更に改良

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

$$P(J|E) = \sum_{a} P(J, a|E) \tag{10}$$

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．

3.1 model1

式 (3) は以下の式に置き換えられる．

$$P(j,a|E) = P(m|E)\prod_{j=1}^{m} P(a_j|a_1^{j-1}, j_1^{j-1}, m, E)P(j_j|a_1^{j}, j_1^{j-1}, m, E) \tag{11}$$

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．

$$P(J,a|E) = \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m} t(j_j|e_{a_j}) \tag{12}$$

$$P(J|E) = \frac{\epsilon}{(l+1)^m}\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\prod_{j=1}^{m} t(j_j|e_{a_j}) \tag{13}$$

$$= \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m}\sum_{i=0}^{l} t(j_j|e_i) \tag{14}$$

model1 において，翻訳確率 $t(j|e)$ の初期値が 0 でない場合， EM アルゴリズムを用いて最適解を推定する．EM アルゴリズムの手順を以下に示す．

手順 1 $t(j|e)$ に初期値を設定する．

手順 2 日本語と英語の対訳文 $(J^{(s)}, E^{(s)})(1 \le s \le S)$ 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．

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

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

$$\lambda_e = \sum_j \sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)}) \tag{16}$$

- 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

$$t(j|e) = \lambda_e^{-1} \sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)}) \tag{17}$$

$$= \frac{\sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)})}{\sum_j \sum_{s=1}^{S} c(j|e; J^{(s)}, E^{(s)})} \tag{18}$$

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．

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．

$$a(a_j|j, m, l) \equiv P(a_j|a_1^{j-1}, j_1^{j-1}, m, l) \tag{19}$$

よって，model1 の式 (6) は以下のように置き換えられる．

$$P(J|E) = \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(j_j|e_{a_j}) a(a_j|j, m, l) \tag{20}$$

$$= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(j_j|e_i) a(i|j, m, l) \tag{21}$$

model2 において，対訳文中の英単語 e と日単語 j が対応付けされる回数の期待値である $c(j|e; J^{(s)}, E^{(s)})$ と，日単語の位置 j と英単語の位置 i が対応付けられる回数の期待値 $c(i|j, m, l; J^{(s)}, E^{(s)})$ が存在する．以下に，期待値 $c(j|e; J^{(s)}, E^{(s)})$ と $c(i|j, m, l; J^{(s)}, E^{(s)})$ を求める式を示す．

$$c(j|e; J^{(s)}, E^{(s)}) = \sum_{j=1}^{m} \sum_{i=0}^{l} \frac{t(j|e) a(i|j, m, l) \delta(j, j_j) \delta(e, e_i)}{t(j|e_0) a(0|j, m, l) + \cdots + t(j|e_l) a(l|j, m, l)} \tag{22}$$

$$c(i|j, m, l; J^{(s)}, E^{(s)}) = \frac{t(j_j|e_i) a(i|j, m, l)}{t(j_j|e_0) a(0|j, m, l) + \cdots + t(j_j|e_l) a(l|j, m, l)} \tag{23}$$

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．

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．

$$P(\phi_0|\phi_1^l, e) = \left(\begin{array}{c} \phi_1 + \cdots + \phi_l \\ \phi_0 \end{array} \right) p_0^{\phi_1+\cdots+\phi_l-\phi_0} p_1^{\phi_0} \tag{24}$$

したがって，model3 は以下の式によって表される．

$$\begin{aligned} P(j|e) &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} P(j,a|e) && (25) \\ &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \left(\begin{array}{c} m-\phi_0 \\ \phi_0 \end{array} \right) p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i! n(\phi_i|e_i) \times \\ &\qquad \prod_{j=1}^{m} t(j_j|e_{a_j}) d(j|a_j,m,l) && (26) \end{aligned}$$

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．

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

$$P(\Pi_{[i]1} = j|\pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_1(j - \odot_{i-1}|\mathcal{A}(e_{[i-1]}), \mathcal{B}(j_j)) \tag{27}$$

- それ以外の場合

$$P(\Pi_{[i]k} = j|{\pi_{[i]1}}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, E) = d_{>1}(j - \pi_{[i]k-1}|\mathcal{B}(j_j)) \tag{28}$$

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．

3.6 GIZA++

GIZA++[?] とは，日英方向と英日方向の対訳文から最尤な単語対応を得るための計算を行うツールである．IBM 翻訳モデルの model1 から model5 に基づいて，単語の対応関係の確率値を計算する．GIZA++を用いた場合，以下の 2 つのファイルが出力される．

1. **T TABLE (Translation Table)** T TABLE は，Model1 から Model3 により作成された翻訳確率 $P(f|e)$ のデータである．f は翻訳する言語で，e は目的言語である．T TABLE は各行が，目的言語の単語 ID(e_id)，翻訳する言語の単語 ID(f_id)，翻訳する言語の単語から目的言語の単語へ翻訳する確率 ($P(f_id|e_id)$) で構成される．

2. **N TABLE (Fertility Table)** N TABLE は，目的言語の単語における繁殖数を表したデータである．N TABLE は各行が，目的言語の単語 ID(e_id)，繁殖数が 0 である確率 ($p0$)，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．

4　先行研究

4.1　一般的な NMT における低頻度語

一般的なニューラル機械翻訳において，計算量削減の目的で語彙数を制限する手法が用いられる [?][?]．この際，使用する語彙数を対訳学習文中における頻度が上位 30,000 語～80,000 語程度とし，それ以外の低頻度語は同一の特殊記号に置き換えられる．しかし，特殊記号は意味を持たない記号であるため，システムに低頻度語が入力された場合の正しい出力が学習されないという問題がある．

4.2　関連研究

NMT における低頻度語の問題を扱う研究は複数存在する．Luong ら [?] は対訳学習文中の単語アライメントの頻度によって対訳単語辞書を構築する手法を提案している．この手法では，Attention 確率を使用せずに対訳学習文の低頻度語を置き換える記号を操作することによって出力文中の未知語を入力文中の原言語単語と対応付けている．これにより，0.48 ポイントの BLEU スコアの向上が確認されている．

関沢ら [?] は低頻度語を同義な高頻度語に言い換える低頻度語処理の手法を提案している．これは前処理として対訳学習文中の低頻度語を言い換え辞書を用いて同義の高頻度語に置き換える前処理を行う手法である．これにより，0.08 ポイントの BLEU スコアの向上が確認されている．

5　提案手法

NMT において，学習時に低頻度語を含む全語彙を用いて学習を行った場合，システム全体の翻訳精度が低下する可能性がある．この理由として，低頻度語は学習の確率値の統計的信頼性が低いこと，さらに対訳学習文中の語彙の多くを占めることが挙げられる．そこで，提案手法では学習時に低頻度語の中でも特に統計的信頼性が低いと考えられる頻度 1 単語を，特殊記号 <unk> に置き換える方法を用いる．この方法を用いた場合，低頻度語を含む文の翻訳において，精度が低下する，あるいは出力文中に <unk> が生成される可能性がある．提案手法ではこの問題を解決するため，対訳学習文と IBM Model 1 を用いて学習した対訳単語辞書を用いる．出力文中の <unk> を，Jean ら [?] の手法により入力文中の原言語単語 (以下，未知語) に置き換え，川原ら [?] の手法により対訳単語辞書を用いて未知語を翻訳する．以上の方法により，翻訳精度の向上を試みる．

5.1　提案手法 (学習)

提案手法における学習の過程を図??に示す．提案手法における学習の過程では，対訳学習文から Attention 確率 [?] の重みを求めてニューラルネットワークに基づくモデル (以下，NN モデル) を学習する．この際，語彙数制限処理として，対訳学習文の日本語文および英語文に含まれる頻度 1 単語を <unk> に置換する．これにより，モデルの学習する語彙数を頻度 1 単語を除いた語彙数に制限する．また，対訳学習文と IBM Model 1 を用いて対訳単語確率を求め，自動的に対訳単語辞書を作成する．

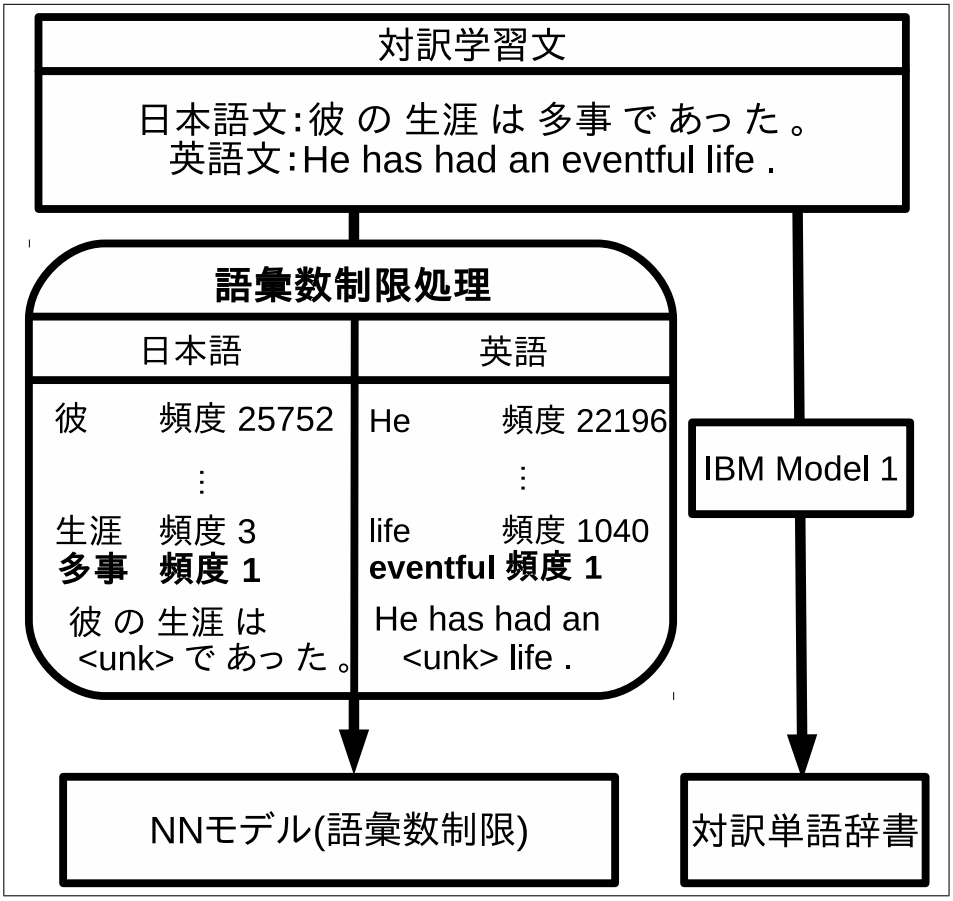

図 1: 提案手法における NN モデルおよび対訳単語辞書の学習

5.2 提案手法 (翻訳および未知語処理)

提案手法における翻訳および未知語処理の過程を図??に示す．翻訳および未知語処理の過程ではまず，学習の過程で得られた NN モデルを用いて入力文から出力文を生成する．NN モデルは語彙数が制限されているため，翻訳時，入力文に低頻度語や未知語が含まれる場合などに，出力文中に <unk> が生成される．<unk> を含む出力文には，<unk> の原言語単語への置換処理 [?] を行う．

最後に未知語処理として，入力文中の原言語単語 (未知語) を含む出力文に対して，未知語を対訳単語辞書から検索し，対訳単語確率最大となる訳語を選択する [?]．訳語の選択が成功した場合は，その訳語を元の出力文の未知語部分に置換し，最終的な出力文を生成する．

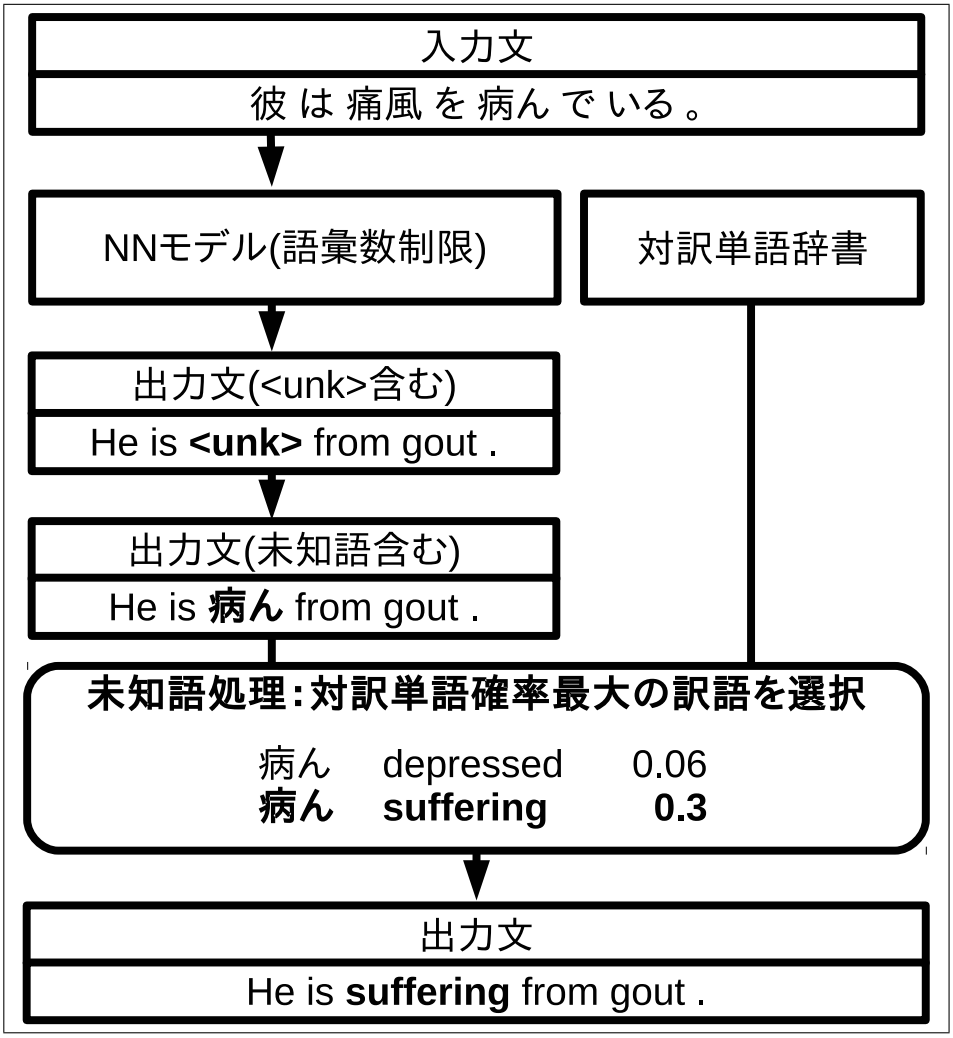

図 2: 提案手法における翻訳および未知語処理

6　実験環境

6.1　実験データ

本研究では，実験に単文のみを用いる．表??に，本研究で用いる単文コーパスの例を示す．

表 6.1: 単文コーパスの例

日本語文	彼 は 自分 の グラス に ウイスキー を つい だ 。
英語文	He poured some whiskey into his own glass .
日本語文	やっと 目的 地 に たどり着い た 。
英語文	I found my way to the destination at last .
日本語文	我々 は 前後 を 敵 に 囲ま れ た 。
英語文	We were surrounded by the enemy before and behind .

本研究では，電子辞書などの例文より抽出した単文コーパス [?] を用いる．使用するデータの内訳を表??に示す．

表 6.2: 実験データ

日本語学習文	160,000 文
英語学習文	160,000 文
ディベロップメント文	1,000 文
テスト文	1,000 文

コーパスの前処理として，各コーパスの日本語文に対して，“MeCab[?]”を用いて形態素解析を行う．また，英語文に対して “tokenizer.perl[?]”を用いて分かち書きを行う．

6.2 評価方法

本研究では，低頻度語を含む全語彙を学習する方法をベースラインとし，ベースラインと提案手法について翻訳精度を比較評価する．翻訳精度の評価として，人手評価を行う．人手評価は，正確性 (adequacy:入力文の意味をどれだけ正確に翻訳英文から読み取れるか) に基づき，対比較評価を行う．ベースライン VS 提案手法の対比較評価では，“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．

6.3 実験設定

NMT のツールキットには OpenNMT [?] を用い，モデルは Luong ら [?] により提案された Global Attention を用いる．Encoder，Decoder の LSTM は 2 層とし, ユニット数は 500，単語の分散表現のベクトルサイズは 500 を設定する．ミニバッチサイズは 40 とし，モデルの訓練は最大 32 エポック行う．Optimizer には SGD を使用し，学習率の初期値は 1 とする．また，各エポックごとに得られたモデルを用いてディベロップメント文を翻訳し，BLEU 値が最高となるモデルを使用する．

7　実験結果

7.1　語彙数

提案手法とベースラインの学習時の異なり語彙数の比較を表??に示す.

表 7.1: ベースラインと提案手法の学習時の語彙数

	語彙数 (日)	語彙数 (英)
提案手法	28,412	25,839
ベースライン	42,760	45,637

ベースラインの語彙数は対訳学習文の異なり語彙数と同様となる．提案手法では，頻度 1 単語処理として対訳学習文における日本語文及び英語文のそれぞれについて，頻度 1 単語を <unk> 記号に置換する処理を行ったため，語彙数が減少している.

7.2　未知語を含む文数

テスト文 1,000 文を入力文として翻訳実験を行い，提案手法の未知語処理前，提案手法の未知語処理後，およびベースラインにおける未知語の数の調査を行った．調査結果を表??に示す.

表??中の “提案手法” とは，本研究で提案した手法の結果である．また，“ベースライン” とは，低頻度語を含む全語彙を学習した場合のモデルを用いた結果である.

表 7.2: 提案手法とベースラインの未知語を含む文数

提案手法 (未知語処理前)	213 文
提案手法 (未知語処理後)	82 文
ベースライン	0 文

表??の結果より，提案手法において，131 文の未知語を削減できたことが確認できる.

7.3　計算量への影響

本研究の提案手法において用いている対訳学習文中の単語を特殊記号に置き換える手法は，従来の NMT において計算量削減の目的で提案されている [?][?][?]．これに関して，頻度 1 単語を <unk> に置換する処理による計算時間への影響を調査した．提案手法とベースラインのモデル学習時の計算時間の違いを比較する．比較結果を表??に示す．

表 7.3: 学習時の計算時間の比較

提案手法	ベースライン
5 時間 5 分 27 秒	7 時間 19 分 44 秒

表??より，提案手法では頻度 1 単語を特殊記号に置換し，学習の語彙数を制限したために計算時間が大きく減少していることが確認できる．

7.4　提案手法におけるシステム全体の翻訳精度

提案手法におけるシステム全体の翻訳精度を調べるために，ベースラインと提案手法の出力文の翻訳品質を比較した．比較方法として，人手評価と自動評価を行った．なお，人手評価には対比較評価を用いる．

7.4.1　人手評価結果

人手評価では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 文を用いる．100 文に対して正確性に基づき，人手による対比較評価を行う．評価の基準を以下に示す．評価結果を表**??**に示す．

表 7.4: ベースライン VS 提案手法における判断基準

提案手法〇	提案手法の方が正確性が高い
ベースライン〇	提案手法の方が正確性が高い
差なし	翻訳精度に明確な差がない

表 7.5: ベースライン VS 提案手法の対比較評価結果 (100 文中)

提案手法〇	ベースライン〇	差なし
36 文	18 文	46 文

表**??**の結果より，人手評価において，提案手法による翻訳精度の向上が確認できた．

ベースライン VS 提案手法における，提案手法○の出力例を表??～表??，ベースライン○の出力例を表??～表??，差なしの場合の出力例を表??～表??に示す．

(a) 提案手法○の出力例

表??において，入力文の “貿易国” に対する出力が提案手法では “country of trade” と比較的正しい訳が得られている．ベースラインでは “American country” となり誤った訳が生成されている．

表 7.6: 提案手法○の出力例 1

入力文	日本 は 貿易 国 で ある 。
参照文	Japan is a trading nation .
提案手法○	Japan is a country of trade .
ベースライン	Japan is an American country .

表??において，入力文の “旅行の申し込みを受け付けています” に対する出力が提案手法では “receiving applications for the trip” と比較的正しい訳が得られている．ベースラインでは “content with my trip” となり誤った訳が生成されている．

表 7.7: 提案手法○の出力例 2

入力文	スペイン 旅行 の 申し込み を 受け付け て い ます 。
参照文	We have opened the books for a tour of Spain .
提案手法○	I am receiving applications for the trip to Spain .
ベースライン	I am content with my trip to Spain .

表??において，入力文の “台所用品” に対する出力が提案手法では “kitchen goods” と比較的正しい訳が得られている．ベースラインでは “kitchen modes” となり誤った訳が生成されている．

表 7.8: 提案手法○の出力例 3

入力文	当社 は 各種 台所 用品 を 供給 し て い ます 。
参照文	We supply all sorts of kitchen utensils .
提案手法○	We supply a variety of kitchen goods .
ベースライン	We supply the various kitchen modes .

(b) ベースライン○の出力例

表??において，入力文の “苦味がある” に対する出力が提案手法では “tastes sweet ” と誤った訳になっている．ベースラインでは，“ tastes bitter” と入力の意味により近い訳が得られている．

表 7.9: ベースライン○の出力例 1

入力文	この チョコレート は 苦味 が ある 。
参照文	This chocolate has a bitter taste .
提案手法	This chocolate tastes sweet .
ベースライン○	This chocolate tastes bitter .

表??において，提案手法では入力文中の “飢饉の年” を正しく翻訳できていない．ベースラインでは全体として比較的正しい翻訳となっている．

表 7.10: ベースライン○の出力例 2

入力文	飢饉 の 年 に は 彼ら は 飢え て しまう 。
参照文	During famine years they starve .
提案手法	They grow hungry in age .
ベースライン○	They go hungry in the age of famine .

表??において，入力文の “救出された” に対する出力が提案手法では “ were displaced” と誤った訳になっている．ベースラインでは，“ were saved” と入力の意味により近い訳が得られている．

表 7.11: ベースライン○の出力例 3

入力文	生存 者 は ヘリコプター で 救出 さ れ た 。
参照文	The survivors were rescued by helicopter .
提案手法	Survivors were displaced by helicopter .
ベースライン○	Survivors were saved by helicopter .

(c) 差なしの出力例

表??において，入力文の “時計” が提案手法では “watch” と訳され, ベースラインでは “clock” と訳されている違いがあるが, 文全体の翻訳精度に差はないと考えられる.

表 7.12: 差なしの出力例 1

入力文	この 時計 は 1 日 に 2 分 遅れる 。
参照文	This watch loses 2 minutes a day .
提案手法	This watch loses two minutes a day .
ベースライン	This clock loses two minutes a day .

表??において，提案手法とベースラインでは同一の出力が得られている．よってこれらを差なしとしている．

表 7.13: 差なしの出力例 2

入力文	私 も 泳げ ませ ん 。
参照文	I , too , cannot swim .
提案手法	I can’t swim .
ベースライン	I can’t swim .

表??において，提案手法では “に落ち着いた” に対する出力 “settled themselves in” が得られ，ベースラインでは “settled into” が得られているが，どちらも同程度に正確であると考えたため，差なしとしている．

表 7.14: 差なしの出力例 3

入力文	彼ら は 新居 に 落ち着い た 。
参照文	They became settled in their new home .
提案手法	They settled themselves in their new house .
ベースライン	They settled into a new house .

7.4.2 自動評価結果

ベースラインと提案手法の出力文（1,000 文）に対して自動評価を行った．表**??**に，自動評価結果を示す．

表 7.15: ベースライン VS 提案手法の自動評価結果. 精度が高い方を太字で示す

翻訳手法	BLEU	METEOR	RIBES	TER
提案手法	**0.2018**	**0.4894**	**0.7866**	**0.5793**
ベースライン	0.1937	0.4756	0.7813	0.5953

表**??**より，実験に用いた全ての自動評価指標において提案手法がベースラインより高い翻訳精度となった．

8 考察

8.1 未知語を含まない文

提案手法の未知語処理前の出力文において，未知語を含まない文は 787 文あった. これらの文について，提案手法○およびベースライン○の場合の例を示す.

(a) 提案手法○の出力例

表**??**において，入力文の “考察している” に対する出力が提案手法では “is considering” と比較的正しい翻訳になっている．しかしベースラインでは “has revolutionized” となっており，誤った訳語が出力されている．この原因として，ベースラインの低頻度語を含む全語彙を学習する方法では，モデルが学習する語彙数が大きすぎるために，出力単語候補の確率的選定の精度が低下していることが推察される．

表 8.1: 未知語を含まない文 (提案手法○の出力例)

入力文	本書 は きわめて 重大 な いくつ か の 問題 を 考察 し て いる 。
参照文	The book discusses some vital issues .
提案手法○	This book is considering several serious questions .
ベースライン	This book has revolutionized several important problems .

(b) ベースライン○の出力例

表**??**において，提案手法の出力文では入力文の意味を大きく損失しているのに対し，ベースラインの出力文では比較的正確な訳となっている．この例では入力文の “状勢” が頻度 1 単語であり，頻度 1 単語を学習しているベースラインでは正しい訳が出力されたと考えられる．提案手法の出力文に入力文中の原言語単語 (未知語) が出力されず，対訳単語辞書を用いた翻訳が行われなかったことも誤りの一因と考えられる．

表 8.2: 未知語を含まない文 (ベースライン○の出力例)

入力文	これ で 状勢 は すっかり 変わっ た 。
参照文	This has completely altered the situation .
提案手法	This is quite changed .
ベースライン○	This transformed the state of affairs .

8.2　未知語処理を行った文

提案手法の未知語処理前の出力文において，未知語を含む文は 213 文あった．そのうち，未知語処理の実行に成功した文は 131 文あった．これら 131 文の翻訳精度について考察する．

8.2.1　人手評価結果

提案手法の未知語処理の実行に成功した文 131 文とベースラインの出力文との人手による対比較評価を行った．評価基準は表??に示したものと同様である．表??に結果を示す．

表 8.3: 提案手法とベースラインの評価結果 (131 文中)

提案手法○	ベースライン○	差なし
56 文	30 文	45 文

表??より，未知語処理の実行に成功した文においても，ベースラインと比較して提案手法の方が翻訳精度が高いという結果になった．これより，提案手法の未知語処理が，翻訳精度の向上に比較的有効であることがわかる．

8.2.2 提案手法〇の出力文の例

また，未知語処理の実行に成功した文のうち，提案手法〇とした例を表??に示す．この例では，入力文中の単語 “香辛料” が低頻度語 (頻度 3) である．提案手法の未知語処理前の出力文において，“香辛料” が未知語として出力されており，最終的な出力文では対訳単語辞書を用いた未知語処理によって “ spices” と置換されている．未知語処理後の出力文は，文法的な正しさを多少欠いているが入力文の意味を概ね推測できる文と考えられる．ベースラインの出力文では “香辛料のきいた食べ物” を “sacred food” と翻訳している．文法的には提案手法よりも正しいが，入力文の意味と大きく異なっている．

表 8.4: 未知語処理を行った文:提案手法〇の出力例

入力文	香辛料 の きい た 食べ物 が 好き だ 。
参照文	I love spicy food .
提案手法 (未知語処理前)	I like the 香辛料 food .
提案手法〇 (未知語処理後)	I like the spices food .
ベースライン	I like the sacred food .

この例のように，出力文中の未知語を文法的に妥当な位置に生成した上で，対訳単語辞書を用いた未知語処理において比較的正しい訳語を選択できた例が，表??における提案手法〇の結果 56 文中 51 文あった．またこの例のように，提案手法では出力文の文法性 (あるいは流暢性) がベースラインより多少低いものの，正確性が高い出力文が得られる傾向にあった．

8.3 対訳単語辞書の問題

表??において，対訳単語辞書の問題により提案手法よりもベースラインの方が良いとされた例が存在した．その例を表??に示す．ベースラインは “歌謡コンテスト” を “music contest”，提案手法は “entry contest” と訳しており，ベースラインの方が入力の意味により近いと評価できる．

この例から，対訳単語辞書の精度の問題が指摘できる．提案手法で用いている対訳単語辞書において，“歌謡” に対する訳語が “entry” と誤った対応になっている．

表 8.5: 提案手法において対訳単語辞書の問題を含む文

入力文	歌謡 コンテスト で わたし は １０ 万 円 の 賞金 を もらっ た 。
参照文	I won one hundred thousand yen in prize money in the singing contest .
提案手法 (未知語処理前)	I won a prize of 100,000 yen in the 歌謡 contest .
提案手法 (未知語処理後)	I won a prize of 100,000 yen in the entry contest .
ベースライン○	In the music contest I won a prize of 100,000 yen .

提案手法において，このような対訳単語辞書の誤りが出力文全体の翻訳精度の低下を招いている例は 131 文中 16 文存在する．これら 16 文のうち 15 文は表??においてベースライン○もしくは差なしと評価されている．

このことから，提案手法において対訳単語辞書の精度が一つの課題であると考察できる．提案手法の対訳単語辞書は対訳学習文と IBM Model 1 を用いて作成しており，IBM Model 1 の精度の問題により対訳単語辞書中に不適切な対訳単語が含まれていると考えられる．これは人手で作成した対訳単語辞書を利用することや，対訳単語確率を得る手法を変更すること (例えばIBM Model 2 を用いる，など) で改善できる可能性がある．対訳単語辞書の精度が改善できれば，表??のような例において，未知語処理後の出力文の翻訳精度向上が期待できる．

8.4　入力文中の単語の頻度と精度に関する調査

提案手法およびベースラインの出力文の翻訳精度について，入力文中の単語の対訳学習文における頻度の影響を調査した．

8.4.1　頻度 1〜10 の単語を含む入力文

入力文中に低頻度語が含まれる場合の，提案手法およびベースラインの翻訳精度について調べた．“低頻度” とすべき頻度を定義することは困難であるが，ここでは低頻度語を含む入力文として，対訳学習文中における頻度が 1 以上 10 以下となる単語を含む入力文とする．ただし，このうち対訳学習文中に出現しない単語を含む文は除外する．このような文は 1,000 文中 392 文存在した．これら 392 文における自動評価結果を表??に示す．

表 8.6: 低頻度語を含む入力文に対する出力文の自動評価結果

翻訳手法	BLEU	METEOR	RIBES	TER
提案手法	**0.1919**	**0.4695**	**0.7710**	**0.6063**
ベースライン	0.1713	0.4387	0.7604	0.6324

表??の結果より，ベースラインと比較して，低頻度語を含む文の翻訳精度向上に提案手法が有効であることがわかる．

8.4.2　頻度 1〜10 の単語を含まない入力文

第??項において対訳学習文中における頻度が 1 以上 10 以下となる単語を含む入力文を低頻度語を含む入力文として評価した．これに対して，低頻度語を含まない文として対訳学習文中における頻度が 11 以上となる単語のみを含む入力文とする．ただし，このうち対訳学習文中に出現しない単語を含む文は除外する．このような文は 1,000 文中 524 文存在した．これら 524 文における自動評価結果を表??に示す．

表 8.7: 低頻度語を含まない入力文に対する出力文の自動評価結果

翻訳手法	BLEU	METEOR	RIBES	TER
提案手法	**0.2307**	0.5296	**0.8051**	**0.5393**
ベースライン	0.2304	**0.5312**	0.8046	0.5455

表??の結果より，ベースラインと提案手法において，低頻度語を含まない文における翻訳精度にはほとんど差がないことがわかる．

9　おわりに

本研究では NMT における低頻度語の問題に着目し，翻訳精度向上を目的とする低頻度語処理の手法を提案した．NMT におけるモデルの学習時，低頻度語を含む全語彙を学習した際に，システム全体の翻訳精度が低下するという可能性がある．この理由として，低頻度語は学習の確率値の統計的信頼性が低いこと，さらに対訳学習文中に多く出現することが考えられる．

そこで本研究では，NMT における新たな低頻度語処理の手法を提案し，システム全体の翻訳精度の向上を試みた．本研究の提案手法では，まず学習時に，語彙数制限処理として対訳学習文中に 1 回のみ出現する単語を全て特殊記号 (<unk>) に置換し，学習を行う．次に翻訳時，Jean ら [?] による <unk> の置換処理を行い，出力文中の <unk> を Attention 確率が最も高い原言語単語に置き換える．最後に未知語処理として出力文に含まれる未知語を対訳学習文と IBM Model 1 により作成した対訳単語辞書を用いて置換する [?]．結果として，出力文 100 文における人手評価では提案手法〇が 36 文，提案手法×が 18 文となった．これより，提案手法を用いた場合，ベースラインと比較して翻訳精度が向上することが確認できる．また，表??，表??より，本研究の実験結果は，低頻度語の学習が NMT のシステム全体の翻訳精度の低下を招くという仮説を裏付ける結果となっていることがわかる．

謝辞

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

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”, *In Proceedings of ICLR*, 2015.

[2] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. “On using very large target vocabulary for neural machine translation”, *In ICML*, 2014.

[3] 川原宰, 村上仁一. “日英翻訳における IBM Model 1 を用いた未知語処理”, 言語処理学会 第 24 回年次大会, 2018.

[4] KyungHyun Cho, “ Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, 2014.

[5] 関沢祐樹, 梶原智之, 小町守. “目的言語の低頻度語の高頻度語への言い換えによるニューラル機械翻訳の改善”, 言語処理学会 第 23 回年次大会, pp. 982–985, 2017.

[6] Ilya Sutskever, Oriol Vinyals, Quoc V. Le “Sequence to Sequence Learning with Neural Networks ”, 2014.

[7] 川原宰, 村上仁一. “統計翻訳における未知語処理 ”, 平成 27 年度 卒業論文, pp. 4-9, 2016.

[8] 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.

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

[10] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective approaches to attention-based neural machine translation ”, *In Proceedings of EMNLP*, pp. 1412–1421, 2015.

[11] 村上仁一, 藤波進 “日本語と英語の対訳文対の収集と著作権の考察 ”, 第一回コーパス日本語学ワークショップ, pp.119-130. 2012.

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

[13] 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.

[14] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. "Opennmt: Open-source toolkit for neural machine translation", 2017.

[15] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies", In S. C. Kremer and J. F. Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, 2001.

[16] 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.