

## 近代文語体と現代口語体の自動翻訳への試み

林英里香<sup>†1</sup> 竹本有紀<sup>†1</sup> 石川由羽<sup>†2</sup> 高田雅美<sup>†1</sup> 城和貴<sup>†1</sup>

**概要:** 本稿では、近代文語体と現代口語体の翻訳に向け、ニューラル機械翻訳を用いて実験を行う。日本語の文語体は各時代で異なり、文語体に関する専門知識がなければ容易に読むことができない。そのため、より多くの人が歴史的資料から知識を得るには、異時代間での自動翻訳が必要である。歴史的資料は、テキストデータがほとんど存在しないが、本稿では、テキスト化されている書籍が比較的多い、近代文語で書かれた文学作品を対象として翻訳を試みる。ニューラル機械翻訳の一つである Encoder-Decoder モデルを機械翻訳の手法として用いて実験を行う。Encoder-Decoder モデルのうち、LSTM 層をもつモデルと畳み込み層をもつモデルの2つを用いて、比較する。

**キーワード:** ニューラル機械翻訳, 畳み込みニューラルネットワーク, リカレントニューラルネットワーク, Seq2Seq

### An attempt to translation of Early-Modern Japanese literal style into modern colloquial Japanese

ERIKA HAYASHI<sup>†1</sup> YUKI TAKEMOTO<sup>†1</sup> YU ISHIKAWA<sup>†2</sup>  
MASAMI TAKATA<sup>†1</sup> KAZUKI JOE<sup>†1</sup>

#### 1. はじめに

日本語で書かれた文章は、時代により文体が異なる。例えば、平安時代の仮名文、それを模して書かれた雅文体、鎌倉時代から明治時代の言文一致運動が起こるまで使用された候文がある。比較的現代語に近い近代以降の文語体に限っても、明治時代の言文一致運動や戦後に行われた国語の大改革などによる違いがある。また、公文書や文学作品のように分野の違いによっても文体の差がある。近代は、西洋文化を吸収し、文明や制度を新たに導入し始めた時代である。書物にも影響が顕著に表れており、新しく作成された単語が定着することなく、以降使用されない低頻出語となる表現が多く存在する。そのため、容易に読むことができる人は限られる。このような特徴をもつ近代文語で書かれた歴史的資料を、多くの人が利用可能にするためには、異なる時代間での翻訳技術が必要である。膨大な量の歴史的資料をすべて手動で翻訳することは困難であるため、自動的に翻訳する手段として、異言語間の翻訳に用いられている機械翻訳を用いる。

機械翻訳には、以前から用いられていたルールベース翻訳[1]や統計的機械翻訳[2][3]があるが、本稿では、ニューラル機械翻訳を用いる[4][5]。近代文語体は、多様な文体で、かつ、低頻出語が多い。そのため、細かいルールを与えずに、良好な精度で自然な文章を生成可能なニューラル機械翻訳が適していると考えられるためである。ニューラル機械翻訳を行うには、適切な形に整えられた大規模な対訳コ

ーパスが必要である。異言語間で自然な文章を翻訳するには、学習データとして数十から数百万文程度の、大規模な対訳文が必要である。異言語間では、インターネットの普及や SNS によって、必要な量の対訳文を得ることが可能となった。しかし、大量に存在する近代書籍は、ほとんどが活版印刷の画像データであり、テキストデータ化されていない。さらに現代語訳された書籍も非常に少なく、対訳文を得ることは困難である。こうしたなかで、テキスト化された対訳データが比較的多いドメインは文学作品である。そのため、本稿では明治時代中期の森鷗外の書籍からデータを数千文収集し、翻訳モデルの構築を行う。ニューラル翻訳では多言語の同時翻訳の際、翻訳する複数の言語すべてに対訳データは必要ではない。例えば、英語と日本語、韓国語の同時翻訳を行う場合は、英日、日英、英韓、韓英のコーパスが十分にあれば、日本語と韓国語間の対訳コーパスが少ない場合も、また、存在しない場合も翻訳が可能である[6][7]。対訳の少ない小規模対訳コーパスであれば、そのコーパスを用いて個別で翻訳モデルを生成するよりも、様々な言語の対訳コーパスと同時に学習させて翻訳モデルを生成する方が、高い精度で翻訳できる。これにより、小規模対訳コーパスでのデータ量の問題は軽減されると考えられる。今後、他分野、他時代の対訳コーパスを用いて翻訳を行うとき、使用した森鷗外の文語体コーパスとの同時翻訳により、あらゆる近代文語体から現代口語への翻訳が可能になると考えられる。他分野の対訳コーパスとして、百文程度の小規模データではあるが、今回公文書データを用いても実験を行う。本稿では、ニューラル機械翻訳の一つである、Encoder-Decoder 翻訳モデルを用いて翻訳を試みる。LSTM を用いるモデルと、CNN を用いる conv2S2S

<sup>†1</sup> 奈良女子大学 Nara Womens University  
<sup>†2</sup> 滋賀大学 Shiga University

(convoluntinal Sequence to Sequence)を近代文語体と現代口語体の翻訳に適用し、比較を行う。

本稿では、2章でニューラル機械翻訳について、Encoder-Decoder モデル、convS2S を述べる。3章では、2章で述べた手法を用いて実験を行う。

## 2. ニューラル機械翻訳

ニューラル機械翻訳は、ニューラルネットワークを用いる機械翻訳の手法である。翻訳元言語を入力系列として与えた際、対訳尤度を最大化するよう学習し、目的言語に翻訳する。ニューラル機械翻訳の1つである Encoder-Decoder モデルは、入力処理するエンコーダと、出力を生成するデコーダを組み合わせたモデルである。エンコーダとデコーダそれぞれに LSTM(Long Short-term memory)を用いるモデルと、CNN を用いた convS2S について以下で述べる。

### 2.1 Encoder-Decoder モデル

Encoder-Decoder モデルは、可変長な入力文をもとにして出力文を生成するモデルである。基本的なモデルは、入力処理するエンコーダと、出力を生成するデコーダとの2つの RNN からなる。デコーダは、エンコーダの出力を参照して、一つ前の隠れ状態ベクトルと、一つ前の予測結果を入力とする RNN であり、次単語の予測を繰り返して目的言語の単語列を生成する。エンコーダに入力文を入力し、文末記号の go タグが入力されると、デコーダでは、文の生成を終える文末記号 EOS(End Of Sentence)タグを生起するまで出力する。RNN では長期記憶が困難であるため、LSTM や GRU を用いる手法もある[4][5][6][7]。

Encoder-Decoder モデルでパラメータを変えず計算コストを上げない状態で、入出力のシーケンスを長くすると翻訳精度が落ちる。長文の対応を可能にするため、Bahdanau らによって Attention を用いたニューラル機械翻訳モデルが考案されている[4]。Attention モデルは、エンコーダの各ステップの隠れ状態のベクトルを記憶し、単語の対応をデコーダで参照して計算する手法である。

### 2.2 convS2S

convS2S は、英仏間と英独間において、LSTM を用いたニューラル機械翻訳に比べて、翻訳のスコアが高い。また、畳み込み層と、ゲート機構付きの活性化関数(Gated Linerar Unit, GLU)を用いている。そのため、並列計算が可能となり高速化できる[8][9]。

図2で示す convS2S の処理手順は以下の通りである。

- 手順I トークン化したテキストを埋め込み
- 手順II 畳み込みして GLU へ
- 手順III Multi step attention を計算
- 手順IV Attention を参照して予測

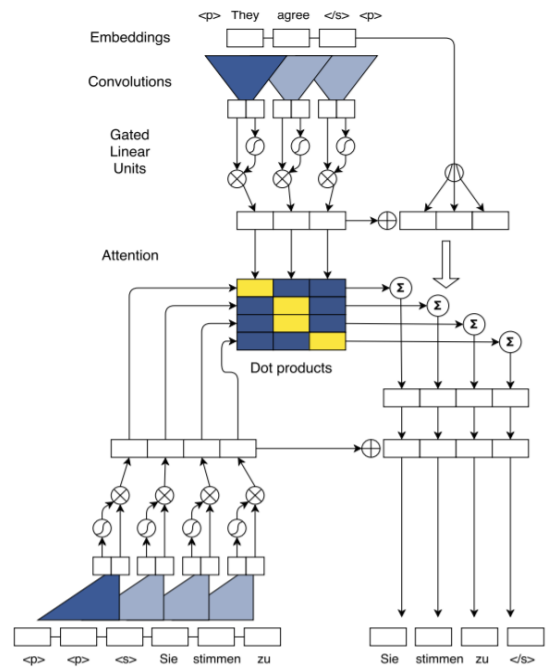


図1.Convolutional Sequence to Sequence の概略図  
 (Jonas[8]の figure1 より転載)

手順 I では、単語列の位置情報をモデルに与えるため、位置埋め込みを行う。入力系列を埋め込んだ行列に、その行列に対する位置埋め込み行列を加える。エンコーダ、デコーダともに同じ処理を行う。手順 II では、まず、入力ベクトル  $X \in \mathbf{R}^{k \times d}$  を重み  $W \in \mathbf{R}^{2d \times kd}$ 、 $b_w \in \mathbf{R}^{2d}$  のカーネルで畳み込み、 $Y \in \mathbf{R}^{2d}$  とおく。GLU 層では、畳み込み層  $Y$  を2つに分岐し、 $[A \ B] \in \mathbf{R}^{2d}$  とし、以下の変換を行う。

$$v([A \ B]) = A \otimes \sigma(B) \quad (1)$$

ここで、 $A, B \in \mathbf{R}^d$  はそれぞれ非線形な畳み込み層のカーネルであり、 $\otimes$  は要素ごとの積、 $\sigma$  はシグモイド関数を表し、出力  $v([A \ B]) \in \mathbf{R}^d$  は、 $Y$  の半分のサイズである。 $d$  は次元数、 $k$  はカーネルサイズである。深層での処理を可能にするため、各隠れ層では残差接続を行う。また、この処理は言語モデルのため、カーネルが生成されるべき目的語の単語を参照して予測前のモデルに影響を与えないよう、畳み込み層の入力を後方へシフトし、先頭から  $k-1$  の長さを0でパディングする。

手順 III では、すべての層で、個別に Attention を再計算する。これにより、過去の履歴の参照が可能になる。 $l$  層目のデコーダの出力を  $h_l^i$ 、一つ前の予測単語の埋め込みベクトルを  $g_i$  とおき、 $l$  層目の  $i$  番目  $d_l^i$  を以下のように表す。

$$d_l^i = W_d^l h_l^i + b_d^l + g_i \quad (2)$$

$W_d^l$  は重み行列、 $b_d^l$  はバイアスであり、 $h_l^i$  はデコーダの出力である。デコーダでの Attention の  $a_{ij}^l$  は、エンコーダの最終ブロック  $u$  の  $j$  番目の出力  $z_j^u$  を用いると次のようになる。

$$a_{ij}^l = \frac{\exp(d_i^l \cdot z_j^u)}{\sum_{t=1}^{T_x} \exp(d_i \cdot z_t)} \quad (3)$$

文脈の式は、位置埋め込みベクトル  $e_j$  を用いて以下で表す。

$$c_i^l = \sum_{j=1}^{T_x} a_{ij}^l (z_j^u + e_j) \quad (4)$$

手順 IV では、Attention で得る入力とデコーダの文脈情報から、単語を生成する。

### 3. 実験と考察

本稿では、近代文語体と現代口語体の機械翻訳に向け、LSTM を用いた Encoder-Decoder モデルと、LSTM の代用として CNN を用いたモデルで、翻訳を試みた。

#### 3.1 実験設定

##### 3.1.1 実験データ

今回の実験では、文学作品と公文書の対訳データを使用する。文学作品は、近代文語体で書かれた森鷗外の文学作品である「即興詩人」のデータを用いる。近代文語体のデータは、青空文庫の「即興詩人」(図書カード:NO.4376)、現代語のデータは、神西清著の作品を用いる[10][11][12]。対象データは、森鷗外、神西清のどちらの作品もハンス・クリスチャン・アンデルセンの「即興詩人」を翻訳したものであるため、それぞれ意識で表現が異なり、1対1の対訳データではない。ニューラル機械翻訳では大規模の対訳データが存在すれば、意識の特徴にも対応できると考えられるが、近代文語体と現代口語体の対訳データはほぼ存在しない。そのため、今回は、どちらかにしか書かれていない表現部分は除いた。また、モデルには Attention を用いているため日本語同士の翻訳であっても、長文に対応できると考えられるが、上記で述べたように大規模データは存在せず、今回は非常に小規模なデータであるため、長文では十分なデータ数が集められず、学習できない。そのため、長文の文章を一部読点で区切ったものを、実験用データとする。他分野の対訳データである公文書の対訳データは、『一般社団法人 近現代史データバンク』[13]より、『文官任用令及文官試験規則ヲ定ム』、『新日本建設ニ関スル詔書』、『太陽暦頒行ノ詔』の3つを使用する。近現代史データバンクでは公文書を現代文に直訳した文の画像データと、現代文のテキストデータを提供している。画像データであるが、文のフォントは統一されているため、OCR を用いてテキストデータにする。

機械翻訳に関する研究の対象は、英語やフランス語のようにスペース区切りがある言語であることが多い。一方、日本語は、文中にスペースが挿入されることはなく、単語の境界の判別が困難である。そのため、実験用データは形態素解析でのトークン化が必要となる。本稿では、形態素解析エンジン MeCab を適用する[14]。MeCab の解析モデルは

bi-gram マルコフモデル、パラメータ推定の学習モデルは CRF(Conditional Random Fields)である。辞書として、近代文語のデータには近代文語 UniDic、現代語のデータには UniDic を用いる[15][16]。

##### 3.1.2 環境

文学作品の実験用データは、全 2048 文、データの配分は、訓練用に 1880 文、テスト用に 100 文、評価用に 68 文を用いる。意識部分を消さず、長文の文章をそのまま用いるデータも使用したデータでも実験を行う。このデータは、全 7892 文、訓練用に 7174 文、テスト 650 文、評価用に 68 文である。これを整理前データとする。公文書の対訳データは、全 181 文である。このデータを他時代他分野の同時翻訳が可能か検証するため用いる。Melvin ら[7]は、実験データの文頭にトークンを付けて多言語同時翻訳を行っている。そこで、この公文書の対訳データにトークンを付ける。目的言語を現代語とするとき <2ge>、明治の文書とするとき <2me> を文頭につけたものを実験データとする。現代語に向けた翻訳と、明治の文語体に向けた翻訳にトークンをつけるので、全 362 文となる。訓練用 290 文、テスト 40 文、評価 32 文とする。

実行 OS は、Ubuntu16.04、使用する計算機は、Intel® Xeon® CPU E5-2620 v4@2.10GHz であり、GPU は NVIDIA GeForce GTX 1080 を 4 枚用いる。

また Encoder-Decoder モデルは、TensorFlow の sequence to sequence のチュートリアルをもとに実装を行った[17]。LSTM を用いたモデルは、エンコーダ、デコーダに各 2 層の LSTM、隠れ層のサイズは 256 次元、バッチサイズ 4、学習率の初期値は 0.5 で確率的勾配降下法を用いて学習を行った。convS2S を用いたモデルでは、エンコーダ側に 4 層、デコーダ側に 3 層の CNN、カーネルサイズは 3 とした。隠れ層のサイズ等のパラメータは LSTM のモデルと同値を適用した。どちらのモデルも、評価用データのデコード時にビーム幅 5 のビーム探索を行って翻訳する。

#### 3.2 実験結果と考察

まず、整理前データを用いて LSTM モデルで実験を行う。データ数別に、情報理論における平均分岐数であるパープレキシティの変化を求める。全データの中から訓練データを 50 文、500 文、5000 文使用したものと。全データを使用した実験結果を図 2 に示す。パープレキシティの収束はデータが少ないほど速いが、すべて 1 に収束している。出力結果は、いずれのデータ数でも以下に示す、文学作品の学習初期段階の結果(表 6)と同様の出力である。数千程度のデータ数の実験では、長文や意識は十分に学習できないとわかる。翻訳精度の評価指標として、パープレキシティのみを用いると、今回のような小規模データでは、どのようなパラメータを用いて実験を行っても、学習が十

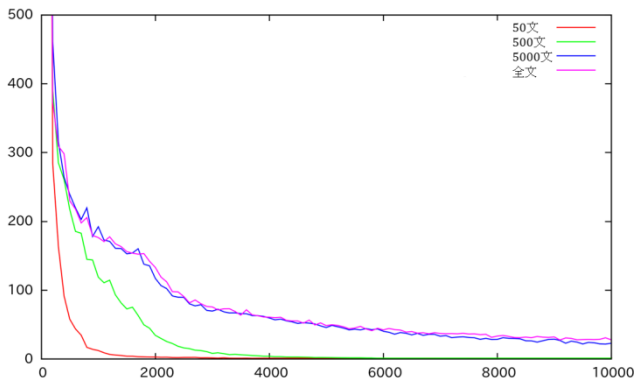


図2 LSTMモデルを用いたデータ数別のパープレキシティの変化(整理前データ)

表1 LSTMとconvS2SでのBLEU, 処理時間の比較

手法	層数		BLEU	処理時間 (s/100epoch)
	エンコーダ	デコーダ		
LSTM (整理前)	2	2	4.6	46.86
LSTM	2	2	71.82	31.35
convS2S	2	2	53.92	2.78
convS2S	3	4	<b>76.09</b>	3.82

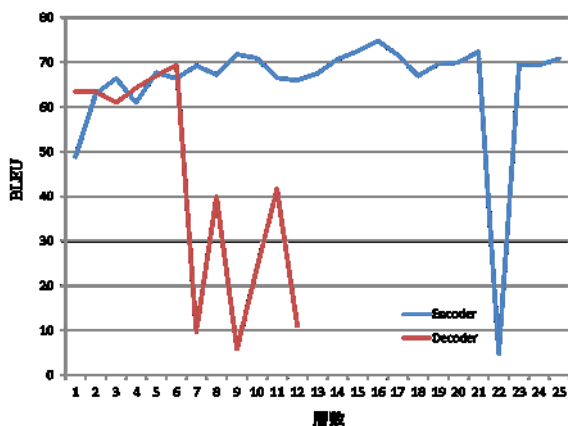


図3 エンコーダ, デコーダの層数変更によるBLEU変化

分に進む前に1に収束するため比較ができない. そのため, 評価指標にはBLEUを用いる[18].

次に整理後の文学作品のデータを用いて実験を行う. LSTMモデルとconvS2Sモデルを用いて, 評価用の対訳データセット68文を翻訳した際のBLEUスコアと, 学習回数100エポックごとの平均処理時間を表1に示す(1万エポックまで). 整理前データを用いた際の結果も示す. 学習回数は, パープレキシティが収束し学習が終了したと考えられるまで訓練を行い, それぞれ10万エポックで学習を打ち切った. 以下に述べる各実験も10万エポックのモデルで比

表2 初期学習率0.5におけるエンコーダ, デコーダの層数変更によるBLEUの変化

カーネル サイズ	エンコーダの層数			デコーダの層数		
	5	9	13	3	5	7
3	61.93	65.53	71.09	<b>76.09</b>	64.95	8.76
5	68.23	69.58	66.65	66.2	68.96	7.76
7	65.1	<b>73.14</b>	67.85	69.07	66.44	7.5

表3 初期学習率0.25におけるエンコーダ, デコーダの層数変更によるBLEUの変化

カーネル サイズ	エンコーダの層数			デコーダの層数		
	5	9	13	3	5	7
3	60.5	39.1	40.78	59.06	39.36	50.89
5	59.03	60.97	<b>76.02</b>	60.14	61.97	<b>73.64</b>
7	60.13	63.52	71.03	12.05	7.11	44.46

表4 初期学習率0.75におけるエンコーダ, デコーダの層数変更でのBLEUの変化

カーネル サイズ	エンコーダの層数			デコーダの層数		
	5	9	13	3	5	7
3	69.78	77.02	59.03	<b>76.84</b>	75.95	76.47
5	75.71	<b>78.68</b>	73.31	74.53	74.58	53.28
7	74.97	75	75.12	73.95	74.59	70.84

表5 最適化関数変更によるBLEUの変化

最適化関数	BLEU
Adagrad	77.23
RMSprop	26.73
ftrl	20.24
Adam	9.79
確率的勾配降下法	<b>76.09</b>

較を行う. まず, convS2SモデルをLSTMモデルのネットワーク構造に合わせ, エンコーダ, デコーダともに畳み込み層を2層として実験を行う. 表1より, convS2Sモデルでの翻訳の方がBLEUスコアは低いが, 100エポックあたりの学習にかかる処理時間は約10分の1である. CNNを用いることで計算の並列化が可能になり, 高速化されると分かる. convS2Sモデルでのカーネルサイズを4, 5に変更して実験を行うと, 各BLEUスコアは59.23, 55.28となり, いずれも表1に示すLSTMモデルのスコアには及ばなかった. 次に, LSTMモデルをconvS2Sモデルのネットワーク構造に合わせたモデルは, 上手く収束しなかった. 3.1章で示したLSTMモデルと, convS2SモデルのBLEU値を比較すると, convS2Sモデルでの翻訳の方が約4ポイント高いという結果となった. 本稿の評価尺度に用いたBLEU値は, 語順や単語が大きく異なる英語とドイツ語間

表 6 100 エポック時の出力結果  
 (入力文は即興詩人[12] 23 頁 29 行目, 29 頁 21 行目より引用)

入力文	出力文
「上手なこと！」と母が言うと、	さえ さえ 同じ 同じ 同じ 同じ 同じ
君だって、たしかに少年詩人だよ！」	とき 光 光 光 光 光 けれど けれど けれど けれど けれど けれど けれど

表 7 10 万エポック時の出力結果  
 (入力文は即興詩人[12] 37 頁 9~11 行目より引用)

入力文	出力文 (正解文)
はいつて来やしないから。	思ひ切つて来ぬものは、 (入るものにはあらず。)
よく寝るんだよ、可愛い坊や！」	熟寐せよ。」 (神の子と共に熟寐せよ。)」)
これだけ言って、	斯く云ひ畢りて、 (斯く云ひ畢りて、)
彼は門をかけてしまった。	をちは戸を鎖ぢて去りぬ。 (をちは戸を鎖ぢて去りぬ。)

表 8 10 万エポック時の出力結果(公文書データ)

入力文	出力文
<2ge>第十八條 文官普通試験は各官廳の須要に應じ其の廳の文官普通試験委員之を行ふ	第八條 文官試験は毎年一回東京に、文官高等試験委員が実施する。
<2me>第二十一条 本令は明治二十七年一月一日より施行する。	第二條 條文官は別に任用の規程を設くるものの外左の資格の一を有する者の中より之を任用す

でニューラル機械翻訳を行う場合 30 前後、英語とドイツ語間よりは語圏の近い英語とフランス語で 40 前後である[8]。日本語同士の翻訳は文法、単語など共通する部分が多い。非常に小規模なコーパスであっても異言語間のニューラル機械翻訳に比べるとスコアは高くなると思われるため、70 前後というスコアは妥当であると考えられる。

convS2S モデルの BLEU スコアは、エンコーダとデコーダの層数が各 2 層の際と、3 層と 4 層にした際で約 20 ポイント異なる。これは、ネットワーク構造に影響を受けていると考えられる。そのため、エンコーダ、デコーダの層数と畳み込み時のカーネルサイズなどを変更し、実験を行った結果を図 2、表 2 に示す。実験した層数とカーネルサイズは、Jonas(2017)[8]の図 2、表 7、8 を参考にしている。表 2 ではエンコーダ、デコーダのそれぞれの層数でカーネルサイズを変更して生成したモデルでの BLEU を求め、一番高いスコアを太字にしている。図 2 から、エンコーダは 16 層付近までは深層になるほど精度が高く、デコーダは 5、6 層付近が高くなる事が分かる。表 2 から、エンコーダ側は、エンコーダの層数を 9 層、カーネルサイズを 7 に変更して生成したモデルが一番高く、デコーダ側では、変更な

しのモデルが一番高いと分かる。カーネルサイズの差異により、相関が特に見られなかったため、初期値を 0.25, 0.75 としたモデルで同じ実験を行った結果を表 2, 3 で示す。初期値のみ異なるモデルを生成することでは、はっきりとした相関はとれなかったが、表 1, 2, 3 からカーネルサイズが 5 のとき、比較的高いスコアが出ると分かった。

また、10 万エポックで打ち切ることで学習が終了しきっていない可能性もあるが、初期値により BLEU のバラつきがあるため、最適化関数の変更を行ったモデルの比較実験も行った。最適化関数には、Adagrad[19], Adam[20], ftrl[21], RMSprop[22]を用いた。結果は表 4 に示す。Adagrad で生成したモデルが変更なしの確率的勾配降下法を用いた際よりも 1.14 ポイント高いスコアとなった。

次に本稿で一番 BLEU の高い構造である、エンコーダを 7 層に変更した際の 100 エポック時の出力結果の一部を表 5、10 万エポック時の出力結果の一部を表 6、表 7 に示す。表 7 には括弧内に正解文も示す。表 6 の出力文は、学習初期段階のため文として成立しておらず、ニューラル機械翻訳で頻繁にある、同じ単語を繰り返し生成する現象が起こっている。学習語の出力結果を表す表 7 で、表の下から 2 文

は正解文と完全に一致している。1 文目の出力文は正解文と異なるが、入力文の「来」と「ない」部分から「来ぬ」と学習できており、似た意味の文を出力できていると思われる。2 文目は、正解文にある「神の子と共に」という部分は予測できていない。しかし、入力文にその部分の記述はなく、文学作品特有の意識表現部分と捉えられるため、入力文に対しては翻訳できていると思われる。そのため、完全に森鷗外の文語体表現を学習できてはいないが、2000 文程度の非常に小規模なデータセットであっても、日本語同士の対訳であれば、ある程度意味の通じる文章には翻訳できているといえる。

表 8 に示す、公文書データでの結果では、1 文目は、現代語の文末表現や意味を翻訳できており、数字以外は概ね同じ意味である。2 文目は、翻訳はほとんどできていないが、「第〇條」の表現や文末は、明治文語体の表現である。使用する対訳データが数百文であっても、「第〇条」など、よく用いられる単語や文末表現は翻訳できている。文頭のトークンによって同時翻訳ができているとわかる。時代だけでなく分野のトークンもつけることで、他時代他分野の同時翻訳が可能であると考えられる。

## 4. まとめ

本稿では、日本語の異なる時代間の翻訳として比較的現代語に近い近代文語と現代語の翻訳を目指している。従来のルールベース機械翻訳や統計的機械翻訳では、低頻出語を多く含む近代文語には適さない。したがって、対訳データをもとに、文全体から直接モデルを作成するニューラル機械翻訳を用いて実験を行い、結果を示す。ニューラル機械翻訳の手法のうち、Encoder-Decoder モデルを用いて、ネットワーク構造内に LSTM 層を含むモデルと、畳み込み層を含む convS2S モデルを比較した。CNN を用いる convS2S モデルでは、エンコーダとデコーダの層数やカーネルサイズ、最適化関数を変更して実験を行った。convS2S の実行処理時間は、LSTM のモデルの約 10 分の 1 であり、CNN の並列処理によって計算が高速化できている。

評価尺度に用いた BLEU スコアは、エンコーダとデコーダそれぞれ 2 層の LSTM を用いるモデルでは 71.82 であった。初期学習率 0.75、エンコーダに 9 層、デコーダに 3 層、カーネルサイズを 5 とした convS2S モデルの BLEU スコアは 78.68 である。本実験で一番スコアが高く、今回用いた森鷗外の文語体作品 2048 文の対訳データセットに適しているといえる。

エンコーダは 16 層付近までは深いほど精度が良く、デコーダは 5, 6 層付近、カーネルサイズは 5、最適化関数には Adagrad を用いるモデルで良い結果がみられた。構造によって BLEU スコアに大きくバラつきはあるが、convS2S モデルは非常に高速である。また、小規模な対訳コーパス

を用いても高い精度を得ることができるため、本実験には LSTM よりも convS2S が適している。また、実験データにトークンを付けて実験した結果から、他ドメインと他分野の同時翻訳が可能であるとわかる。

今後は、あらゆるドメイン、また、明治の初期、中期、大正、昭和前期と、時代によって変化する近代文語体にも対応する翻訳モデルを目指し、実験を行う。

## 参考文献

- [1] Sholom M. Weiss et. al. Rule-based Machine Learning Methods for Functional Prediction. Journal of Artificial Intelligence Research 3.1995.383-403p.
- [2] P.F.Brown et. al. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19(2). 1993. 263-311p.
- [3] Phillip Koehn et. al. Statistical phrase-based translation. NAACL HLT vol.1. 2003.48-54p.
- [4] Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. International Conference on Learning Representations. <http://arxiv.org/abs/1409.0473> . 2015.
- [5] I.Sutskever et.al. Sequence to sequence learning with neural networks. Advances in neural information processing systems 27,NIPS.2014.3104-3112p.
- [6] Yonghui Wu et.al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv preprint arXiv:1609.08144v2.2016.
- [7] Melvin Johnson et.al. Google's Multilingual Neural Machine Translation System:Enabling Zero-Shot Translation. arXiv preprint arXiv:1611.04558 .2016.
- [8] Jpnas Gehring et.al. Convolutional Sequence to Sequence Learning. arXiv preprint arXiv:1705.03122v2. 2017.
- [9] YN Dauphin et.al. Language Modeling with Gated Convolutional Networks.arXiv preprint arXiv:1612.08083.2016.
- [10] 青空文庫 <https://www.aozora.gr.jp/cards/000019/card4376.html>(accessed:2018-11-14)
- [11] アンデルセン,神西清.即興詩人(上)(下) kindle 版. 2015-03-04.
- [12] アンデルセン,神西清.即興詩人.角川文庫.1960.
- [13] rekishiru <http://rekishiru.com/>(accessed:2018-11-14)
- [14] MeCab: Yet Another Part-of-Speech and Morphological Analyzer <https://taku910.github.io/mecab/> (accessed:2018-11-14)
- [15] 小木曾智信,小町守,松本裕治.歴史的日本語資料を対象とした形態素解析.自然言語処理,Vol.20 No.5.727-748p.2013.
- [16] Unidic <http://unidic.ninjal.ac.jp/> (accessed:2018-11-14)
- [17] TensorFlow <https://www.tensorflow.org/> (accessed:2018-11-14)
- [18] K.Pascanu,T.Mikolov,and W.J.Zhu.BLEU: a Method for Automatic Evaluation of Machine Translation.ACL 40.2002.311-318p.
- [19] John Duchi,Elad Hazan,Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.MLR 12.2011.2121-2159p.
- [20] DP Kingma,Jimmy Lei Ba.Adam: A Method for Stochastic Optimization.ICLR.2015.
- [21] H.Brendan et.al.Ad Click Prediction: a View from the Trenches. ACM SIGKDD 19th Int. Conf. Knowl. Discovery Data Mining.2013.1222-1230p.
- [22] Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSER: Neural Networks for Machine Learning, 4:2, 2012.