

ウェアラブルデバイスによる曖昧な入力からの ニューラル機械翻訳を用いた日本語文章推定方式

渡辺 淳¹ 原 直¹ 阿部 匡伸¹

概要：近年，スマートウォッチや VR の普及などを背景に様々な文字入力方式が考案されている．そこで本報告では，歩きながらの入力やアイズフリーな入力を可能とするために十指に装着するウェアラブルデバイスを使用し，各指から得られる入力を利用して文章を推定する方式を検討した．検討したシステムでは，ウェアラブルデバイスから各指に対応する 0～9 の 10 種類の数字の系列を入力し，ニューラル機械翻訳を利用することで数字の系列から文章を推定する．評価実験では，約 280 万文の日本語データから独自に作成したパラレルコーパスを学習し，単語正解率 90 % を達成した．

1. はじめに

近年スマートウォッチや VR 技術の発展に伴い，様々な文字入力方式やそのためのデバイスが提案されている．文字入力は，情報検索やコミュニケーションを行う上で不可欠な機能であるが，屋外でコンピュータを使用する場合などでは，通常のフルキーボードによる入力が困難である．そこで本報告では，両手の十指に装着するウェアラブルデバイスの使用を想定し，各指の対応する 0 から 9 の数値のみから，日本語の文章を推定する方式を提案する．本方式を用いることで，アイズフリーな文章入力が，屋外であったり，さらには歩きながらでも可能となる．また，この入力方式を，テキスト音声合成などと組み合わせて，発話に障害を持つ人の会話支援システムに応用することも考えられる [1]．

本研究で提案する少数キーによる入力方式は，欠落した入力情報から期待する情報を推定することが主な目標となる．本稿では，この事を入力の曖昧性と呼ぶ．日本語は平仮名だけでも 50 種，常用漢字を含めばその数は二千を超える．0 から 9 という 10 種類の文字から 2 千種類もの文字を推定しなければならない．

ウェアラブルデバイスによる日本語入力方式として，これまで，スマートウォッチからの入力，指装着型ウェアラブルデバイスからのコマンドを用いた入力，変換モデルを用いた入力などの方式が提案されてきた．

ウェアラブルデバイスの一種であるスマートウォッチで

は，操作画面が小さいため一般的なキーボードを配置することが困難となる．そこで，画面を 4 つなどに分割した文字入力方式が考案されてきた [2], [3]．これも少数キーによる文字入力的一种である．スマートウォッチはタッチパネルを搭載しているため，フリック操作も含めれば操作の幅は広がる．実際，[2] では，画面を 4 分割し，それぞれの区画から 4 方向へのフリック操作を組み合わせることで，入力パターン数を確保している．

また，フルキーボードによる入力が困難な場合の入力デバイスとして，指装着型のウェアラブルデバイスが研究開発されている [4], [5], [6]．これらのデバイスも少数キーによる文字入力を前提としており，いずれもアイズフリーなデバイスの利用が可能となっている．梅舟ら [1] や田中ら [7] は，このような指装着型のデバイスから一意に入力可能なコマンドを作成することで，日本語の入力を可能とした．しかし，このようなコマンド入力方式では，平仮名 50 音のコマンドを記憶する必要があり，システムの習熟にはコストを要する．その他，少数キーによる文字入力の曖昧性の解決には，多くの場合にコマンド方式が利用されてきた [8], [9], [10]．

松原ら [11] は，このようなコマンド方式ではなく，大規模データから変換対を学習し，作成したモデルによって文字入力を行う方式を検討した．また後続の研究として，ニューラルネットワークを用いた方式も検討されている [12]．

本研究では，ウェアラブルデバイスからの入力方式として，タッチタイピングと同じ指の動きで入力可能な入力方式を提案する．さらにタッチタイピングを習得している人，もしくは QWERTY キーボードでの入力経験者を利用

¹ 岡山大学大学院ヘルスシステム統合科学研究科
Graduate School of Interdisciplinary Science and Engineering in Health Systems

者として想定することで、習熟コストを削減することを期待する。

本論文の構成は以下の通りである。2章では、関連研究について述べる。3章では、提案方式について述べる。4章では、データセットの作成と評価実験方法について述べる。5章では、評価実験の結果と考察を述べる。6章では、本論文のまとめと今後の課題について述べる。

2. 関連研究

少数キーからの入力を日本語に変換する方式として、3つの研究を紹介する。

2.1 コマンドによる入力方式

梅舟ら [1] は、発話に障害を持つ人のコミュニケーション支援を目的とし、手袋型の入力デバイスを操作することによりリアルタイムで小型スピーカから音声を出力するシステムを提案した。手袋型入力デバイスには指の先端にシートスイッチが組み込まれており、指先を机などに押すことで、スイッチから信号が送られる。この入力デバイスを両手に装着することで、計 10 個のスイッチによる入力を提案している。

入力方式は、同時入力方式と 2 度入力方式という 2 種類を検討している。いずれの方式も、基本的には右手の 5 本が母音、左手が子音と役割を決めている。日本語の場合、母音の種類は 5 種類だが、子音の種類は 16 種類、拗音等を含めればさらに多くなる。それらの子音を右手の五本指の組み合わせで表現している。同時入力方式では、2 本または 3 本の指の同時押しを駆使して子音を表現する。2 度入力方式では、同じ指の 1 度押しと 2 度押しで別の子音を表現する。

ウェアラブルデバイスを用いている点や、10 個の装置（スイッチ）から日本語を入力するという点で本研究と類似しているが、その変換方式は一意に決定可能な日本語入力方式といえる。そのため、正確な入力が行われた場合には、正しい日本語入力が保証されるが、入力子音の指の組み合わせを覚えるコストは高い。より多くの人に利用してもらうためには、習熟コストの低さは重要な点である。

2.2 帰納的学習によるモデル作成

松原ら [11] は携帯端末の普及を背景に、12 個のキーから日本語入力を行う方式を提案した。携帯端末はその大きさから多数のキーを配置することが困難であり、少数キーから日本語入力を行う必要がある。一般的には、12 個のキーを用いたトグル入力と呼ばれる入力方式が主流であった。トグル入力とは、「い」ならあ行のボタンを 2 回、「せ」ならさ行のボタンを 4 回、と押下回数によって文字入力を行う方式である。トグル入力は利用しやすい単純な入力方式であり、入力の曖昧性も存在しないが、入力に時間がかかる

というデメリットが生じる。例えば「お」であればキーを 5 回押下する必要がある、また、「ぺ」などでは 14 回キーを押下する必要がある。

そこで松原らは、母音を無視した入力方式を提案した。すなわち、あ行→「1」、か行→「2」、… と数字を対応させ、以下のような変換を行う。

わたしは、かれをみた。
0 4 3 6 # 2 9 0 7 4 ##

これによって従来のトグル入力よりも押下回数を削減し、入力速度の改善を図った。

この入力方式は平仮名 1 文字が 1 回のキー入力で可能となるため、フルキーボードによる入力よりも少ない押下回数で日本語入力を行うことができる。また評価実験では、文字単位での変換精度によって評価が行われており、その変換精度は約 85 % であった。しかし、文字単位よりも単語単位での変換精度の方が低く算出されるため、精度が十分であるとは言い難い。本稿では単語単位での変換精度を扱う。

2.3 ニューラルネットワークによるモデル作成

鎌田 [12] らは、[11] と同様に、携帯電話端末の入力を迅速に行うことを目的とし、ニューラルネットワークを用いた少数キーからの日本語入力方式を提案した。

本研究と比較するとこの変換では、1 文字ずつ入力するため、入力文が分かち書きされている必要がない。日本語における自然言語処理ではしばしば文章の分かち書きが障壁となるため、文字単位での解析は優れているといえる。しかし、出力は二字熟語のみであるため、文章を出力するにはさらなるネットワークの拡張が必要となる。

3. 提案方式

3.1 提案方式の概要

提案方式の概要図を図 1 に示す。提案手法は学習部と翻訳部に分かれており、学習部で作成したモデルを翻訳部で用いる。入力された数値列は、ニューラル機械翻訳と学習済みモデルによって日本語仮名漢字交じり文に変換される。学習済みモデルについては、3.3 節で後述するが、数値列と日本語文を対訳データとした学習データを学習したものである。

入力は 0 から 9 の十種類の数値からなる数値列であり、以下のように単語区切りの文単位で行われる。

214128 / 71 / 3184187238 / 3327

(私 / は / 大学生 / です)

ここで、単語とは形態素と同等とする。上記の数値列は「私は大学生です」という文を意味している。数値の羅列は一般的なタッチタイピングを行う際、各指に割り当てられるキーを 0 から 9 の番号としたものである (図 2)。つまり、「私」という単語はキーボードでは「watasi」と入力さ

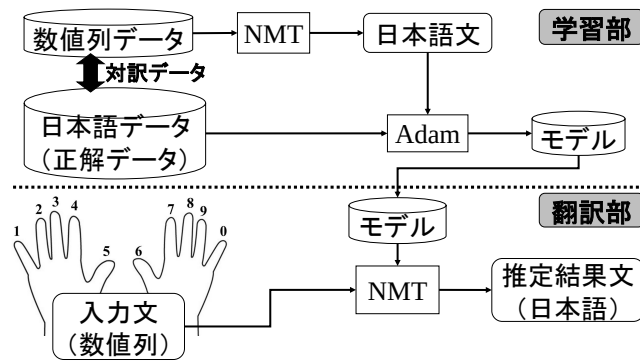


図 1 提案方式の概要図

Fig. 1 Overview of proposal method

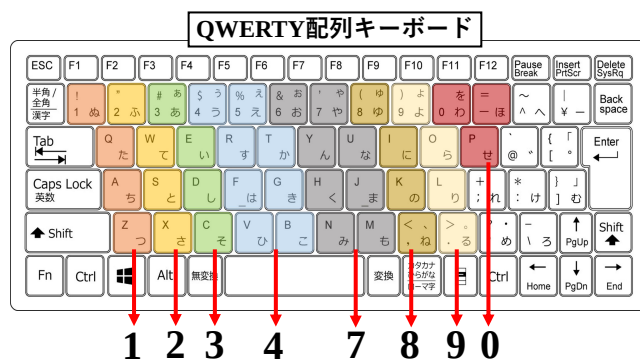


図 2 タッチタイピングにおける各指の割当

Fig. 2 Assignment of each finger in touch typing

れると仮定し、これを図 2 に習い数値に変換することで「214128」という数値列となる。また、単語の区切りは、スペースで示す。即ち、入力は 5 (左の親指) となる。また、文の終わりは、改行で示す。つまり、入力は 6 (右の親指) となる。この時の入力文中の単語間の前後関係を考慮することで、文単位での日本語文章推定を行う。

3.2 変換先の競合問題

提案方式の主な問題点として、入力された数値列に対して変換先の単語が競合する場合があることが挙げられる。これは、アルファベット 26 文字を、0 から 9 の 10 種類の数値で表現するという曖昧性に起因する。つまり、QWERTY キーボードにおける縦列の情報が欠落しているということである。例として、「朝」という単語を意図して「121」と入力した場合を考える。しかしこの場合、「121」の変換先として「泡」という単語の可能性も生じる。このように、入力された数値列が複数の単語に変換され得る場合が数多く発生する。そこで、ニューラル機械翻訳を利用し、文中の単語の前後関係を考慮したモデルを作成することで変換先の競合問題の解決を目指す。

3.3 ニューラル機械翻訳

ニューラル機械翻訳(NMT:Neural Machine Translation)

とは、ニューラルネットワーク (NN:Neural Network) によって自然言語の解析・翻訳を行う、機械翻訳の手法のひとつである。自然言語処理では RNN (Recurrent Neural Network) と呼ばれる NN が一般的に利用される。RNN は時系列データを扱うためのネットワークである。ある文章を単語の順序付きの並び、つまり時系列データと見なし、処理を行う。本研究では RNN の一種である LSTM[13] を用いる。さらに、LSTM を複数連結させた Sequence to Sequence モデル [14] (以下、seq2seq モデルと呼ぶ) によって、対訳データを学習する。

図 3 に提案方式で用いた attention 機構付き seq2seq モデルの概要図を示す。seq2seq モデルは RNN を応用したモデルであり、大きく Encoder と Decoder の 2 つに分けられる。Encoder では元となる文章を処理し、隠れベクトル (Hidden vector) と呼ばれる固定長のベクトルに変換する。Decoder では、Encoder で生成された隠れベクトルから目標とする文章を推定する。さらに、seq2seq モデルに attention 機構 [15], [16] を追加することで、文章を変長ベクトルで扱うことが可能となる。従来の seq2seq モデルは、入力系列における全単語を固定長のベクトルで表現するため、長文になるほど単語の埋め込みが困難になるという欠点があった。attention 機構は、この問題を解決するための機構である。図 3 の赤矢印が attention 機構の重要な点であり、Encoder 側の Hidden vector を全て結合し、Decoder で出力した Hidden vector との内積を求める。得られた内積を基に context vector を生成し、入力文と出力文の単語のアライメントを学習する。

各単語は、語彙データを基に one-hot vector に変換され、さらに Embedding vector に圧縮される (図 4)。Embedding vector にはその単語の分散表現が含まれており、推定時にもこの Embedding vector を参照することで、以下で表される単語の生成確率を求める。

$$P_{\theta}(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{J+1} P_{\theta}(\mathbf{y}_j|\mathbf{Y}_{<j}, \mathbf{X}) \quad (1)$$

ここで $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_I)$ は入力系列、 $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_J)$ は出力系列とする。 $\mathbf{x}_i$ は入力系列 $\mathbf{X}$ の i 番目の単語を表す。

4. 評価実験

4.1 データセットの作成

学習および検証・テストに用いる対訳データの作成について説明する。本報告では、オープンソースの日英対訳データである JESC (Japanese English Subtitle Corpus)[17] を利用する。JESC は日英 (英日) 翻訳タスクのためのコーパスであり、テレビ番組の字幕データによって構成されている。しかし実際には、図 5 のように映画やアニメなどの字幕が大半を占めるため、口語的表現の文章を多く含む。

また、JESC はオープンソースな口語表現コーパスとし

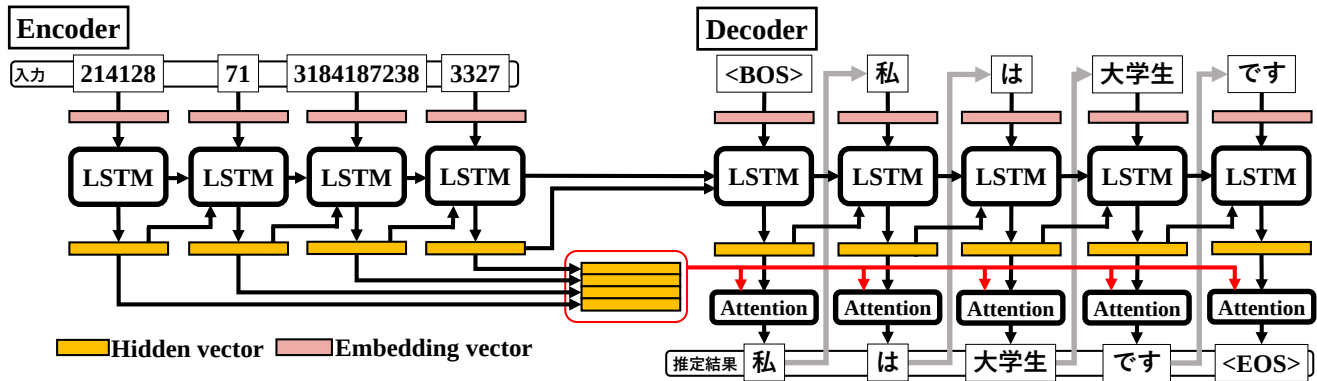


図 3 attention 機構付き seq2seq モデルの概要図. BOS と EOS はそれぞれ Begin Of Sentence と End Of Sentence の略である. BOS は推定する系列の開始を表すと同時に、入力系列の終了を表す. EOS が推定された場合、その直前まで推定されてきた単語群を推定結果文として出力する.

Fig. 3 Schematic diagram of seq2seq model with attention mechanism. BOS and EOS are short for Begin Of Sentence and End Of Sentence, respectively. BOS indicates the start of the sequence to be estimated and the end of the input sequence. When EOS is estimated, the words that have been estimated up to that point are output as the estimation result sentence.

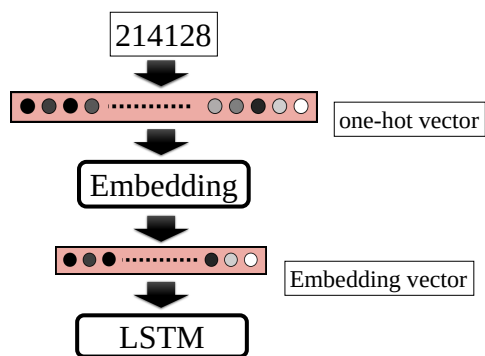


図 4 単語の Embedding
Fig. 4 Embedding of word

ては最大規模のコーパスである. NMT は長文であるほど翻訳精度が低下する事が報告されており、本タスクにおいてもニュース文などのように長文では動作するとは限らない. その点、JESC は口語的な表現が多く、文の長さもあまり長くないと予想されたため、本タスクの最初の試みとしてこのデータベースを利用した.

以下に、データセットの作成手順を示す.

- (1) JESC から日本語文章を抽出
- (2) 句読点、記号等の削除
- (3) MeCab[18] (IPA 辞書) を用いて単語分割
- (4) MeCab によって読み情報 (片仮名列) に変換
- (5) かなローマ字変換に従い、アルファベットに変換
- (6) 図 2 に従い、数値に変換
- (7) (3) と (6) を対訳データとすることでデータセットを作成

we can be in mexico. at least, we can, we can
メキシコに逃げればいい
every girl who died, released her power on to
the next.
全ての少女が 死んで力を解放した 私が最後に残ったとき
i won't deny my own personal desire... to turn
each sin against the sinner.
私自身も喜んで 罪人に罪を償わせた
you can see from here?! no, not from there!
こ... こんな所から見えないのか?
makochan, last year you played the piano to
celebrate...
一応 考えとく。
why don't you believe me? because i'm foreign?
何で信じてくれないの? わたしが 外国人だから?
what is he trying to do?
団長は一体 何を見ようとしているんだ
my teriyaki mayo pizza and ginger ale!
俺様のテリマヨピザとジンジャーエールが
so it uses its red bioluminescence like a
sniper's scope
赤い生物発光を狙撃者の望遠鏡として用いて
i thought maybe you might...
いつもな

図 5 JESC の例文
Fig. 5 example of JESC

4.2 実験条件

4.2.1 各データセットの数

評価実験に用いた各データの文のペア数を表 1 に示す. いずれのデータも JESC から作成した文章であり、それぞれのデータセットは重複する文章を含まない. また、JESC の学習データは元々 280 万文であるが、4.1 の処理を行う際、MeCab の読み解析を行うことができなかった特

表 1 データセットの数
Table 1 A number of Dataset.

学習データ	2,736,262 (ペア)
検証データ	1,965
テストデータ	1,963

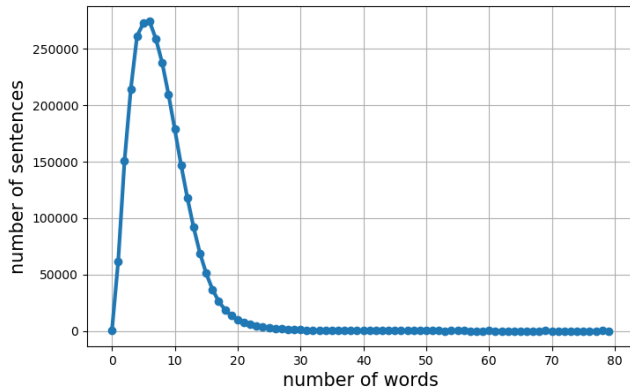


図 6 学習データにおける各文をなす語数の分布

Fig. 6 Distribution of number of word in each sentences about training data

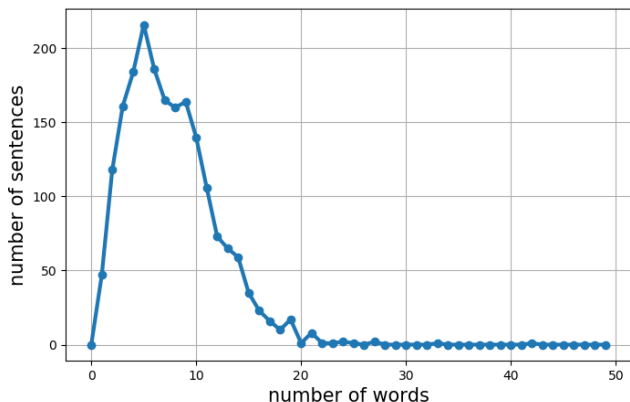


図 7 テストデータに関する各文をなす語数の分布

Fig. 7 Distribution of number of word in each sentences about test data

殊文字などを含む文章をデータセットから除去している。表 1 の各データセットは、全データから特殊文字などを含む約 2% の文章を除去し、それぞれの用途に合わせ分割したものである。

また、学習データとテストデータについて 1 文当たりの語数の分布をそれぞれ図 6、図 7 に示す。いずれも 5 単語から 6 単語からなる文数をピークに分布している。先に述べたように、JESC の字幕データは、口語的な表現が多いため、一文を構成する単語数は比較的少ないものと言える。また、30 単語以上の文は数文しか含まれていないという傾向も、学習データとテストデータで合致している。

4.2.2 学習条件

attention 機構付き seq2seq モデルの学習条件を表 2 に示す。ハイパーパラメータについては、GPU の性能を元

表 2 学習条件

Table 2 training conditions

LSTM ユニット数	1,024
LSTM レイヤ数	3
ミニバッチサイズ	128
最大語数	50
語彙数	50,000
活性化関数	ReLU
損失関数	Softmax Cross Entropy
最適化手法	Adam
使用 GPU	Tesla K40c (Mem:12 GB)

にいくつかの値で試行錯誤し、検証データに対して最も翻訳精度が高くなるよう設定した。

また最大語数とは、提案方式が扱うことのできる一文当たりの最大単語数である。本稿では、データセット中の文を全て扱うことができる 50 単語とした。語彙数は、学習データ中の単語から作成される語彙データのサイズである。これについても GPU の性能（メモリ）を元に決定した。

本稿では、約 10 epoch 学習後に保存したモデルを使用する。

4.3 実験

作成した対訳データを学習し、モデルを作成する。得られたモデルを用いてテストデータを変換し、正解文と比較することで単語正解率を算出する。ある文の総単語数を n_a 、正しく変換できた単語数を n_c とすると、単語正解率は

$$\frac{n_c}{n_a} \times 100 \quad (2)$$

で与えられる。本稿では、文の語数ごとに単語正解率の平均を求め、評価する。

また、自作した別のテストデータによる評価も行い、実際にどのような変換ミスが発生しているかなどを分析する。

5. 結果と考察

5.1 結果

5.1.1 テストデータ

図 8 に語数ごとに単語正解率の平均を算出したグラフを示す。正解率と一緒に図 7 で示したテストデータ数も並記する。単語数 20 以上では、テスト文数が少ない。

テストデータ全文の単語正解率の平均は 92.0% であった。この高い正解率の要因としては、学習・テストにおいて単語数の少ない文を用いたためであると考えられる。一般的に NMT では、長文であるほど翻訳程度が低下することが報告されているが、その長文の目安としては 20 単語以上の文である [15]。そのため 10 単語以下の文を多く含む JESC に対しては、高い正解率を示したと考えられる。

また、語数が 1 などの短文は、提案方式の強みである単語の前後関係の考慮を行うことが困難である。そのため、

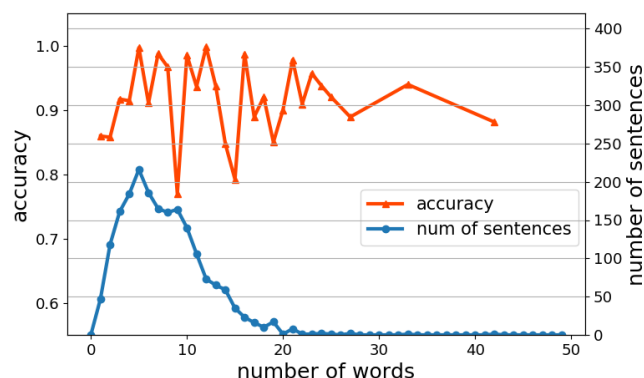


図 8 テストデータの単語正解率

Fig. 8 Accuracy of test data

学習データ中の単語の出現頻度がほぼそのまま反映され、正解率が低くなったと考えられる。

5.1.2 自作データ

以下に自作したテストデータの日本語文を示す。

- (1) 私の名前は渡辺です
- (2) こんにちは
- (3) ここから岡山駅まで何分かかりますか
- (4) この建物はおよそ 80 年前に建立されたものである
- (5) そこの引き出しの中にあるスマホ取ってくれないかな
- (6) 明日の朝までに必ず確認しておいてください
- (7) 3 日後のライブ良ければ一緒に行きましょう
- (8) そんなことできるわけないやん
- (9) ウェアラブルデバイスによる曖昧な入力からのニューラル機械翻訳を用いた日本語文章推定方式
- (10) そこで今回は一文が長い文章と短い文章の具体例を比較し読みやすい文章の書き方について考えてみよう

自作データは固有名詞や口語表現、また文語表現などを含む文によって構成されている。この 10 文を数値に変換し、学習済みモデルを用いて変換した結果を以下に示す。

- (1) 私の名前はワタナベです
- (2) こんにちは
- (3) ここから岡山駅まで何分かかりますか
- (4) この建物はおよそ 80 年前に **<unk>**^{*1}されたものである
- (5) そこの引き出しの中にあるスマホ取ってくれないかな
- (6) 明日の朝までに必ず確認しておいてください
- (7) 3 日後のファイル良ければ一緒に行きましょう
- (8) そんなことできるわけないじゃん
- (9) **<unk>** デバイスによる曖昧な入力からの **<unk>** 機械静脈を用いた日本語文章推定常識

^{*1} unknown の略。語彙データに含まれていない単語であることを表す。

- (10) そこで今回は一文が長い文章と短い文章の具体例を比較し読みやすい文章の書き方について考えてみよう

変換に失敗した単語を太字で表している。推定結果文を見ると、変換に失敗しているケースは主に 2 パターンに分けられる。

一つ目は、語彙データに含まれなかった単語（「ウェアラブル」、「ニューラル」など）、つまり未知語である。語彙データへの単語の登録は、学習データ中の出現頻度順に行われるため、出現回数の極端に少ない単語は登録されない。しかし学習データが有限である以上、未知語の入力は必ず想定されるものであり、モデルは未知語を正しく未知語と判定できる必要がある。例えば文 (4) の「建立^{*2}」という単語は、未知語に変換されているが、この単語は語彙データには含まれないため、未知語を未知語と変換できた例と言える。

二つ目は、変換の曖昧性に起因する失敗である。例えば、「ライブ」と「ファイル」はそれぞれアルファベットでは「raibu」、「fairu」であり、どちらも数値列に変換すると「41847」である。また、「翻訳」と「静脈」、「一文」と「一流」なども同様の失敗である。

5.2 考察

提案方式の問題点である入力の曖昧性について考察する。3.2 でも述べた通り、少数キーからの日本語入力を検討する場合、変換先が競合する問題は避けられない。そこで、曖昧性を持つ単語を 2 つ例に取り、文章に含めて変換を行った。

- 曖昧性を解消できなかった場合
(121, 「朝 (asa)」 「泡 (awa)」)

input: ビールの味は泡が決める
estimated: ビールの味は朝が決める

- 曖昧性を解消できた場合
(8797, 「今日 (kyou)」 「以上 (ijou)」)
input: 今日の授業はこれで以上です
estimated: 今日の授業はこれで以上です

このように、結論としては曖昧性を解消できた単語とできなかった単語が存在する。この原因は、単語の出現回数や他の単語との結び付きの強さ（共起回数）が学習データに依存するためである。例えば、学習データ中の「朝」という単語の出現回数は 1,911 回、今朝などの熟語も合わせれば 5,419 回であるのに対し、「泡」という単語の出現回数は 110 回、熟語を合わせても 266 回と、出現回数に約 20

^{*2} 学習データ中の「建立」の出現回数は 4 回であった。

倍の差がある。さらに、「泡」という単語を含む全ての文章を確認すると、「ビール」と同一の文に出現しているケースは一つも見られなかった。このように、学習するデータセットによって、推定の結果は大きく左右されるものと考えられる。

6. おわりに

本稿では、attention 機構付き seq2seq モデルによって 0 から 9 の数値のみから日本語文章を推定する方式を検討した。評価実験では、テストデータに対する単語正解率の平均値が 92.0%を示した。また、変換に失敗する場合は、未知語によるものと曖昧性によるものがあり、これは学習済みモデルがデータセットに依存するためであると考えられる。

単語単位での変換は高い正解率を達成したため、今後の課題としてはより実用的な入力条件である文節での入力方式を検討する必要がある。本稿では単語の集合を文とするため、入力も単語としたが、MeCab の定める単語を正確に入力するにはある程度の言語的な知識を必要とする。そのため、文節での入力が習熟のコストを削減することに繋がれば、より汎用性の高いシステムを開発することができる。

謝辞

本研究の一部は KDDI 総合研究所との共同研究によるものである。

参考文献

- [1] 梅舟柄安, 大倉典子: 発話障害者のための自然対話支援システムの開発, 横幹連合コンファレンス予稿集第 1 回横幹連合コンファレンス, 横断型基幹科学技術研究団体連合 (横幹連合), pp. 181–181 (2005).
- [2] 安福和史, 中村喜宏: キーの数を最少化したスマートウォッチ向け文字入力方式の提案, 第 80 回全国大会講演論文集, Vol. 2018, No. 1, pp. 359–360 (2018).
- [3] 安福和史, 中村喜宏: QuadKey: キーの数を 4 つに限定したスマートウォッチ向けかな文字入力方式, 情報処理学会論文誌, Vol. 60, No. 8, pp. 1403–1412 (2019).
- [4] : TAP SYSTEMS INC: Tap Strap — The Most Advanced Keyboard In The World(online), <https://www.tapwithus.com/>.
- [5] : 富士通指輪型ウェアラブルデバイス, <https://pr.fujitsu.com/jp/news/2015/01/13.html>.
- [6] 福本雅朗, 外村佳伸: Wireless FingeRing: 人体を信号経路に用いた常装着型キーボード, 情報処理学会論文誌, Vol. 39, No. 5, pp. 1423–1430 (1998).
- [7] 田中純之介, 勝間亮: 指装着型デバイスにおける平仮名入力の効率化, 2019 年度情報処理学会関西支部 支部大会 講演論文集, Vol. 2019 (2019).
- [8] 北村拓郎, 森清人: 少数キーによる文字入力方式, 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI), Vol. 1999, No. 35 (1999-HI-083), pp. 37–42 (1999).
- [9] 杉本正勝: 片手操作キーカード (SHK) による日本語入力, 情報処理学会研究報告モバイルコンピューティングとユビキタス通信 (MBL), Vol. 1997, No. 54 (1997-MBL-001), pp. 1–6 (1997).
- [10] 田中久美子, 犬塚祐介, 武市正人: 少数キーを用いた日本語入力, 情報処理学会論文誌, Vol. 44, No. 2, pp. 433–442 (2003).
- [11] 松原雅文, 荒木健治, 桃内佳雄, 枅内香次: 文字情報縮退方式を用いた帰納的学習によるべた書き文の数字漢字変換手法の有効性について, 電子情報通信学会論文誌 D, Vol. 83, No. 2, pp. 690–702 (2000).
- [12] 鎌田竜也, 松原雅文, 馬淵浩司: ニューラルネットワークを用いた携帯端末向け日本語入力手法における単語変換精度, 第 67 回全国大会講演論文集, Vol. 2005, No. 1, pp. 83–84 (2005).
- [13] Gers, F. A., Schmidhuber, J. and Cummins, F.: Learning to forget: Continual prediction with LSTM (1999).
- [14] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to sequence learning with neural networks, *Advances in neural information processing systems*, pp. 3104–3112 (2014).
- [15] Bahdanau, D., Cho, K. and Bengio, Y.: Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473* (2014).
- [16] Luong, M.-T., Pham, H. and Manning, C. D.: Effective approaches to attention-based neural machine translation, *arXiv preprint arXiv:1508.04025* (2015).
- [17] Pryzant, R., Chung, Y., Jurafsky, D. and Britz, D.: JESC: Japanese-English Subtitle Corpus, *Language Resources and Evaluation Conference (LREC)* (2018).
- [18] 工藤拓, 山本薫, 松本裕治: Conditional Random Fields を用いた日本語形態素解析, 情報処理学会研究報告. NL, 自然言語処理研究会報告, Vol. 161, pp. 89–96 (オンライン), 入手先 (<https://ci.nii.ac.jp/naid/110002911717/>) (2004).