日本語学習者の単語表記の自動評価に向けて

Dolça Tellols[1]　徳永 健伸[1]　横野 光[2]
[1] 東京工業大学 情報理工学院　[2] 株式会社富士通研究所
{tellols.d.aa@m, take@c}.titech.ac.jp　yokono.hikaru@fujitsu.com

1 はじめに

言語学習者の増加にともない，コンピュータを使った言語学習支援 (Computer Assisted Language Learning) システムの研究が関心を集めている [1]. また，学習支援と同時に学習者の言語知識や言語能力を自動的に評定することも重要な研究課題である．文法や語彙の全体的な知識だけではなく，より詳細な要素を測る尺度があれば学習者の言語能力の長所と短所を正確に把握することができる．その結果を元に学習者はより効率的に勉強でき，教員も学習者のニーズに合わせた教え方ができる．

本研究は言語学習者の言語能力を構成する一つの次元である語彙に注目する．語彙は読んだり聞いたりする時に使う理解語彙と書いたり話したりする時に使う使用語彙に分けられると言われている [2, 3, 4, 5, 6]．従来，選択肢問題を使って理解語彙を評定する研究は盛んにおこなわれているが [7]，使用語彙を評定する研究は少ない．その理由のひとつとして，評定対象の語を学習者に使用させる文脈を用意することは難しいことがあげられる．

これまでにも，使用語彙のサイズを予測するテストや尺度がいくつか提案されている．例えば，文を完成させる Productive Vocabulary Levels Test (PVLT) [2]，語彙頻度リストを使って作文に出てくる語彙の分布を測る Lexical Frequency Profile (LFP) [8]，刺激単語から連想する単語を書かせ，その単語の難易度によってスコアを計算する Lex30 テスト [9] などがある．本研究では語彙サイズではなく，単語表記の正確さに着目する．話し言葉の場合，単語表記の正確さを評価するために発音に注目するべきだが，書き言葉の場合，言語の特徴によって注目するべき点が異なる．英語のように一つの表記体系しか使わない言語はこの次元を評価するためにスペルミスに注目することになる．しかし，日本語のように複数の表記体系を持つ言語では，誤字 (スペルミスを含む) だけではなく，使われている表記体系の適切さも考慮するべきである．本稿では，書き言葉において日本語学習者の単語表記の正確さに関する評価次元を定義し，自動評価に向けて解決すべき課題について議論する．

2 関連研究

本研究は日本語学習者の単語表記の正確さを測るために誤字と表記体系の適切さを考慮する．自動評価を目指しているため，キーボード入力を仮定する．キーボード入力における自動タイポ修正を目指している研究はいくつか存在する．Hagiwara ら [10] は Github[1)] データを用いて自動的に多言語スペルミス・コーパス[2)] を構築した．Tanaka ら [11] は Wikipedia[3)] の編集履歴を用いて日本語タイポ・データセット[4)] を構築し，タイプ修正のためにエンコーダーデコーダーモデルを学習した．高橋ら [12] は株式会社リクルートテクノロジーズのデータを使って BLSTM に基づく誤字脱字検出システムを構築した．そして，Komatsu ら [13] は日本語タイポを分類するためにアテンション付き LSTM モデルを学習した．これらの関連研究の多くは母語話者のデータを基にしている．しかし，日本語学習者の誤り傾向は母語話者と違う可能性があるため，実際の学習者から得られたデータも調べる必要がある．

以下，言語学習者データを利用した関連研究を紹介する．Mizumoto ら [14] は母語話者が修正した学習者の Lang-8[5)] データを用いて日本語自動誤字修正のための Lang-8 コーパス[6)] を構築した．その後，Koyama ら [15] はこのコーパスの一部にアノテーションを行なって，エラー修正モデルの評価のために新たなコーパス[7)] を構築し，それを用いて Lang-8

1) https://github.com
2) https://github.com/mhagiwara/github-typo-corpus
3) https://en.wikipedia.org
4) http://nlp.ist.i.kyoto-u.ac.jp/EN/edit.php?JWTD
5) https://lang-8.com
6) https://sites.google.com/site/naistlang8corpora
7) https://forms.gle/roMnZdqd1EKWSM2D9

コーパスで学習したニューラル機械翻訳と統計機械翻訳モデルを評価した．同じデータを用いて，Homma ら [16] も NAR(non-autoregressive) ニューラル機械翻訳モデルを学習し評価している．これらの研究は語彙・文法を併せて修正するモデルを構築しているが，本研究では語彙，特に単語の表記に注目する．

誤字に関する研究が多い一方で，表記体系の使い方を調査した研究は少ない．高校教科書に出てくる字の表記体系の割合を示している研究 [17] や時間経過とともに漢字と仮名の使用率がどう変わるかを調査した研究がある [18, 19]．しかし，その使用率と学習者の習熟レベルとの関係を調査している研究は見あたらない．

Takaoka ら [20] は，日本語の表記体系が複雑で正書法が厳密でないため，表記揺れが日本語形態素解析の課題となっていることを議論している．

3 単語表記の正確さの定義

本研究では日本語の単語表記の正確さを定義するため，入力された語が以下の三つのクラスのいずれかに分類されることを仮定する．

(A) 正解
　誤字を含まず適切な表記体系で書いてある場合．
(B) 不適切な表記体系
　漢字表記が適切だと考えられるのに仮名表記されているなど，使われている表記体系が不適切な場合．
(C) 誤字
　スペルミス，変換ミスなどのように単語として成立していない場合．

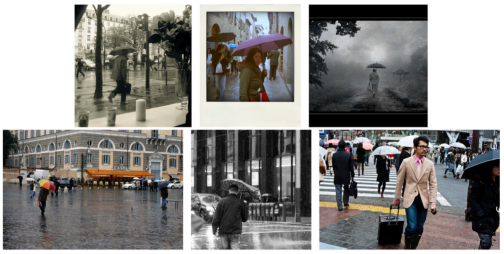
例えば，「今日は共だちとコンサトにいきたいです.」の正しい表記は「今日は友だちとコンサートに行きたいです.」である．この中で「今日」は正しいので (A)，「共だち」は「友だち」の誤字 (変換ミス) なので (C)，「コンサト」は「コンサート」の誤字 (スペルミス) なので (C)，「いきたい」は「行きたい」と書く方が適切なので (B) にそれぞれ分類できる．

4 自動評価に向けての調査

本節では，単語表記の自動評価に向けて日本語学習者と母語話者のデータを分析し，そのデータを分類するモデルを試作し，分類における問題の分析を行った．

4.1 データ

今回分析したデータは東京工業大学に所属している 14 人の日本語学習者と 2 人の日本語母語話者から得られたテキストデータである．学習者集合は様々なレベル (初級者 4 人，中級者 5 人，上級者 3 人，超上級者 2 人)，様々な国籍 (中国 4 人，インドネシア 3 人，台湾 2 人，英国 1 人，サウジアラビア 1 人，米国 1 人，ネパール 1 人，シンガポール 1 人) で構成されている．学習者のレベルは日本語授業のクラス分けテストの結果を基に分類した．



頭に浮かんだ単語：

1. かさ
2. 雨
3. くもり
4. かばん
5. 歩く
6. コート
7. 街
8. 持つ
9. 黒
10. _____ *(無解答)*

画像の説明：

雨が降って色々な人が傘とかばんを持ち、街の中を歩いている。

図 1　データ収集タスクの例

データは STAIR キャプションデータセット [21] を用いて生成した画像セットについて記述するタスク (図 1) から得た．各画像セットの被写体は共通点を持っており，学習者がテキストを書くための刺激になる．学習者には画像セットごとに 3 分以内に頭に浮かんだ 10 語と画像の説明 (可能な限り 20 字以上) を書かせた．図 1 中で青字の部分が学習者の入力に相当する．今回は，10 個の画像セットを使って得られた合計 144 の応答を使っている．このデータは Django[8] で作り，Heroku[9] でデプロイした Web アプリケーションを通して収集した．アプリ内でブラウザーのスペルチェックオートコンプリートとオートコレクト機能を使えないように設定した．

8) https://www.djangoproject.com
9) https://www.heroku.com

4.2 単語表記分類モデルの試作

本研究では Python(バージョン 3.6.0) と以下のライブラリーや API を使って単語表記の分類を実装した.

1. pykakasi[10], バージョン 1.2.
2. Google CGI API for Japanese Input[11].
3. mecab-python3[12], バージョン 0.996.2(デフォルト辞書).

pykakasi は平仮名と片仮名への変換のために使い, Google の API はかな漢字変換のために使った. 日本語では単語の間に区切りがないため, MeCab を用いて単語分割とフィルタリング (名詞-固有名詞と数以外-, 自立動詞, 形容詞と副詞だけを考慮する) を行った.

試作した分類モデルの入力は, 図 1 の課題に対して入力された「頭に浮かんだ単語」と「画像の説明」である. 使っている画像と STAIR キャプションデータセットにある対応しているキャプション (母語話者が書いた) を文脈として扱う. さらに, 語として成立するかどうかを判断するために『現代日本語書き言葉均衡コーパス (BCCWJ)』の語彙表 (BCCWJ 長単位語彙表)[13] を使った.

学習者の入力を 3 節で述べた単語表記のクラスに分類する処理を以下に示す. 前処理として, 入力から空白と改行を削除する.

(1) 入力を Google の API によって変換する.
(2) 4.3.3 節の分析に基づき, 文脈を参照し, 変換結果に以下の後処理を行う:
  – 漢字一文字とその次の部分の変換結果を修正する.
  – 平仮名一文字と漢字一文字の変換を修正する.
  – ある単語の変換結果が複数ある場合, その中に文脈に表れるものがあればそれを採用する.
(3) MeCab を使って (2) の結果に単語分割とフィルタリングを行い, 単語集合を得る.
(4) 元の入力とそれを平仮名に変換したものを参照し, (3) で抽出された各単語を次のように分類する:
  – 単語にローマ字が含まれている場合か, 変換ミスと認識された場合か, 語彙表に存在していない場合は (C) に分類する.
  – 誤字が含まれなくて入力に対して処理された単語の表記体系に変化があった場合は (B) に分類する.
  – それ以外の単語は (A) に分類する.

例えば, 入力として「きれいなそらで　たこをあそびに　そとへいきます。」を与えると, (1)〜(3) の処理の結果として次の単語集合が抽出できる:「きれい」,「空」,「たこ」,「遊び」,「外」,「行き」, 各単語に対して, (4) の処理で「きれい」と「たこ」はクラス (A) に分類され,「空」,「遊び」,「外」と「行き」はクラス (B) に分類される.

4.3 議論

今回の実験は規模も小さく, 手法も暫定的なものなので, これまでにわかった問題点について定性的な分析を述べる.

4.3.1 誤字判断の難しさ

学習者の入力を人手で確認した際, 誤字かどうかの判断が困難な事例が見られた. 例えば,「じゅ」のように平仮名で書かれてあるものは学習者が何について言及したかったのかを推測することが難しく, 定義したクラスのどれに該当するかを判断することができない. 誤字かどうかを判断するためには文脈を考慮することが重要である. 例えば, 電車の写真が提示された問題に対してある学習者が「記者」と書いたとする. これ自体は単語として存在しているため, 正確さには問題ないと判断してしまう. しかし, 文脈を考慮すると恐らく学習者は「汽車」と書きたかったのだろうが, かな変換の誤りで「記者」としてしまったと推測することができる.

また, 語彙の誤りなのか文法の誤りなのかの判断が難しい事例もある. 例えば,「飛びる」のような入力に対して, 学習者は「飛ぶ」か「飛べる」を書きたかったと推測するとこの入力を活用の誤りだとみなして, 語彙の正確さに関しては問題がないと判断することもできる. しかし, 本来は存在しない単語である「飛びる」を書いたと推測するとこの入力は語彙に関する誤りとして単語表記の正確さの評価の際に考慮する必要がある.

「シャレオツ」のような俗語も学習者は入力しており, このような語を誤字とすべきか, 正確な語と

10) https://github.com/miurahr/pykakasi/
11) https://www.google.co.jp/ime/cgiapi.html
12) https://taku910.github.io/mecab/
13) https://pj.ninjal.ac.jp/corpus_center/bccwj/

して扱うべきかを決めなければならない．

試作のモデルでは判断が難しかった誤字もいくつかあった．例えば，「たくらん」(学習者は「たくさん」を書きたかったと推測される) が処理の後に「托，卵」になり誤字として扱えなかった．また，単語分割や変換の問題で判断できなかった誤字には「レフォーム」(「リフォーム」),「みなあさん」(「みんなさん」),「どこ度も」(「どこでも」),「おとも」(「とても」) などがあった．

誤字を正確に認識するためにはどの文字列が誤字であるかがアノテーションされたデータが必要だと考える．そのデータを用いることで単語分割モデルを改良し，誤字を含む単語の認識率を上げることができると考えられる．

4.3.2 表記体系の適切さの判断の困難性

日本語の表記には平仮名，片仮名と漢字があり，日本語話者は状況に応じてそれらを使い分ける．漢字で書ける語彙は漢字で書く場合が多いが，その表現を柔らかくするため平仮名で書いたり，強調するために片仮名で書いたりすることもある．それに加えて，読み手を考えて，文章を読みやすくするために表記体系を調整することもありえる．例を挙げると「うち (家)」,「とまる (止まる)」,「たこ (凧)」,「はは (母)」,「ひも (紐)」,「わかりません (分かりません)」,「ごみ (ゴミ)」などがある．

単語表記を (B) として判定する際に，その単語単独で判断するのではなく，入力全体の表記のバランスも考慮して判定することも考えられる．

4.3.3 Google の API の問題点

API から得られる出力が正しい変換結果でない場合がある．例えば，木が写っている画像に対する単語として入力された「き」が「木」ではなく「気」になり，「むら」が「村」ではなく「ムラ」になり，「よる」は「夜」ではなく「よる」のまま残る，ということが起きる．また，文脈によって API の変換結果も異なる場合がある．例を挙げると，「うち」はそのまま入力すると「家」にならないが「うちもきれいとおもいます.」のような入力においては「家」と変換される．

Google の API が行う単語分割において誤りが発生することもある．例えば，「この動物は白いと黒い線があります.」の入力において，「白いと」は「白，糸」になり，「黒い」は「黒，位」になる．また，「楽しむ活動」は「楽し無活動」になり，「奪うために」は「奪うた目に」になるといった問題もある．この問題に対しては，API の入力に文節区切りの位置を指定することができるため，別の自動単語分割システムの結果に従い，区切りを決めたら API から得られる出力が改良できると考えられる．

4.3.4 MeCab の単語分割の問題点

平仮名の入力に対しては，単語分割が誤る場合がある．例えば，「たべもの」(食べ物) は「たべ，もの」になり，「かぞく」(家族) は「か，ぞ，く」になり，「そと」(外) は「そ，と」になる．

それに加えて，入力に誤字が含まれている場合も，その誤字の影響で単語分割が誤ることがある．例を挙げると，「りょきょうのためにじゅんびします.」(今日のために準備します) という文章を入力すると「りょ」という誤字のために「り，ょきょうのためにじゅんびします,.」という単語分割を得る．これを解決するために現在の試作でかな変換を行なった後で単語分割することを検討している．しかし，今使っているかな変換 API からまだ理想的な出力は得られない．別の解決方法としては，アノテーションされたデータを使い，平仮名で書いてある単語と誤字を含む単語を考慮する辞書を準備し，単語分割システムがその辞書を使うようにすることが考えられる．

5 おわりに

本稿では，日本語学習者と言語教育関係者を支援するために，単語表記の正確さを定義し，自動評価に向けて調査した．単語表記の正確さの定義において誤字と表記体系の適切さに注目し，ルールベースでの単語表記分類モデルを試作した．実際に学習者が作成したデータを用いて評価を行うことで，解決すべき課題や問題点が明らかになった．特に，試作したモデルでは用いたかな変換と単語分割システムの誤りが問題となる．

単語表記の正確さの自動評価に向けて，今後の課題はスコアの決定，試作した分類モデルの改良とアノテーションガイドラインの作成である．

参考文献

[1] Detmar Meurers and Markus Dickinson. Evidence and interpretation in language learning research: Opportunities for collaboration with computational linguistics. *Language Learning*, Vol. 67, No. S1, pp. 66–95, 2017.

[2] Batia Laufer and Paul Nation. A vocabulary-size test of controlled productive ability. *Language testing*, Vol. 16, No. 1, pp. 33–51, 1999.

[3] Stuart Webb. Receptive and productive vocabulary sizes of l2 learners. *Studies in Second language acquisition*, Vol. 30, No. 1, pp. 79–95, 2008.

[4] Birgit Henriksen. Three dimensions of vocabulary development. *Studies in second language acquisition*, Vol. 21, No. 2, pp. 303–317, 1999.

[5] Paul Nation. *Learning vocabulary in another language*. Cambridge University Press, 2001.

[6] John Read. *Assessing Vocabulary*. Cambridge University Press, 2000.

[7] David Beglar and Paul Nation. A vocabulary size test. *The language teacher*, Vol. 31, No. 7, pp. 9–13, 2007.

[8] Batia Laufer and Paul Nation. Vocabulary size and use: Lexical richness in L2 written production. *Applied linguistics*, Vol. 16, No. 3, pp. 307–322, 1995.

[9] Paul Meara and Tess Fitzpatrick. Lex30: An improved method of assessing productive vocabulary in an L2. *System*, Vol. 28, No. 1, pp. 19–30, 2000.

[10] Masato Hagiwara and Masato Mita. Github typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. *arXiv preprint arXiv:1911.12893*, 2019.

[11] Yu Tanaka, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. Building a Japanese typo dataset from wikipedia' s revision history. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 230–236, 2020.

[12] 高橋諒, 蓑田和麻, 舛田明寛, 石川信行. Bidirectional lstm を用いた誤字脱字検出システム. 人工知能学会全国大会論文集 一般社団法人 人工知能学会, pp. 3C4J903–3C4J903. 一般社団法人 人工知能学会, 2019.

[13] Ryuki Komatsu, Rin Hirakawa, Hideaki Kawano, Kenichi Nakashi, and Yoshihisa Nakatoh. Study on mistype correction support using attention in Japanese input. In *2020 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–2. IEEE, 2020.

[14] Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning sns for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 147–155, 2011.

[15] Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 204–211, 2020.

[16] Hiroki Homma and Mamoru Komachi. Non-autoregressive grammatical error correction toward a writing support system. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pp. 1–10, 2020.

[17] Takushi Tanaka. Statistical analysis of Japanese characters. In *COLING 1980 Volume 1: The 8th International Conference on Computational Linguistics*, 1980.

[18] Dilhara Darshana Premaratne. Is the use of kanji increasing in the Japanese writing system? *Electronic Journal of Contemporary Japanese Studies*, 2012.

[19] Yuko Igarashi. *The changing role of katakana in the Japanese writing system*. PhD thesis, 2007.

[20] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: A Japanese tokenizer for business. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[21] Stair captions: Constructing a large-scale Japanese image caption dataset. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 417–421, Vancouver, Canada, July 2017. Association for Computational Linguistics.