1. はじめに

独立行政法人国際交流基金によると、2009年現在、133の国や地域で365万人が日本語を学習している。
日本語学者はそれぞれ異なった文化や、母語の統語・音韻論的な影響など様々な背景がある。
日本語は普通、中国語から取り入れられた表意文字である漢字と、音節を表す文字である仮名を用いて書かれる。
仮名には約50個の異なる文字がある。これらをすべて覚えるのはラテン音圏の日本語学者には負荷が高く、多くの学習者はローマ字を用いて学習を始める。
しかし、日本語母語話者にとっては、仮名で書かれている方がローマ字で書かれているよりも添削しやすい。
そのため、ローマ字を仮名に変換することにより、日本語添削者が楽に添削できるようになるが、学習者の文はスペル誤りを含んでいるため、単純なローマ字仮名変換機能はうまく動作しない。
一般に用いられてきた日本語入力システムはローマ字を仮名に変換する機能を持っているが、これらは日本語を母語とする者を対象にしているため、学習者の誤りを含む入力を正しく扱うことができない。

本稿ではローマ字で書かれた誤りを含むテキストを、誤り訂正された仮名のテキストに変換するタスクを提案する。これにより、日本語添削者が学習者の文を添削する際、文を容易に読むことができるようになる、我々の手法は3つのステップから構成される。
すなわち、言語の特定、スペル訂正、そしてローマ字仮名変換である。以下ではそれぞれについて詳しく説明する。

言語の特定 日本語学習者は日本語には存在しない、自分の母語の単語をそのまま書くことがある。
日本語のローマ字化の方法は、単純にスペルを反映したものではないため、これらの単語はうまく仮名で変換できない。そのため、これらの単語は仮名に変換せずに残しておいたほうが日本語添削者にとっては読みやすい。本稿では英語の辞書を用意し、マッチする単語は仮名に変換せずに残しておく。

スペル訂正 誤りを含む単語は正しく数値できないので、スペル訂正が有効である。

ローマ字仮名変換 前述の方法で訂正された文字列に対し、単純な方法を用いて、ローマ字を仮名に変換する。
実験は言語学習SNSから収集したデータを使用し提案手法で変換した結果と、標準的な日本語入力システムの結果を比較し、正しく変換できなかった場合の誤り傾向を分析し、さ
らに学習者の誤りの特徴を紹介する。
本稿の構成を述べる。まず第２節に係る学習者の誤りの特徴を紹介する。第３節では、日本語のローマ字化について簡単に触れ、第４節で学習者の誤りの特徴を示す。

### 2. 関連研究
我々の主な関心は誤りを含む入力をどう扱うかというところにある。文献7では仮名で書かれた文における誤り検出と訂正を仮名の文字 n-gram 情報を利用して行った。我々の手法もこの方法に似ているが、ローマ字入力に焦点を当てるなら、許容する中国語の入力システムが紹介された。ローマ字仮名変換は、ピクン漢字変換と似ているが、我々の目的は日本語教師を支援しているところに違いがある。文献2では高精度な言語学的手法を紹介した。日本語学習者の誤り訂正は文献3でも試みられている。彼らは仮名漢字変換の文の訂正を試みたが、我々の対象はローマ字で書かれた文である。

### 3. 日本語のローマ字化
日本語のローマ字化についてはいくつか異なる手続きがある。そのうち多く使われているものとして、ヘボン式、訓令式、そして日本式ローマ字がある。ヘボン式は最も広く使われている。ほとんどの日本語学習者もヘボン式を使用しているため、提案手法では変換の手続きとしてこれを使用することにした。ヘボン式は一般的に英語の音読法に基づいている。これは日本語の単語の発音をローマ字によって表現するので、英語話者には習得が容易な方法である。ローマ字化にこれよりもしさまざまな難題が伴うが、ほとんどの難題は長音を示すための記号の有無によるものである。

### 4. Lang-8 から得た学習者のローマ字日本語コーパス
ローマ字で書かれた日本語学習者のコーパスを筆者らが知る限りでは存在しない。そのため、ローマ字の学習者のコーパスを言語学習 SNS の Lang-8よりから集めた。このサービス上では約 75,000 人のアクティブユーザが様々なトピックについて記事を書き込んでいる。また各ユーザ自身が学習したい語をプロフィールに登録しているため、この情報をもとに日本語学習者の文だけを集めた。日本語学習者が書いた文が 925,588 文蓄積されており、そのうち 93.4% に当たる 763,971 文が人手での検討を受ける。さらに約 10,000 文がローマ字で書かれたものである。

表 1 に Lang-8 で見られる文の例を示す。Lang-8 から得られたデータにはそのインターフェースや文化からくる特徴がある。例えば、1 や 2 のように添加後の文の横に、添加者が補足するためやコミュニケーションを取るためにコメントをつけている文がある。また 3 や 4 の文のように、ローマ字の文と漢字の文とを併記している場合もある。5のようにハイフンを複合語の接続に用いている学習者もいる。6 の文では、この文だけではわからない主語を、添削者が前の文脈から推察して補完している。7 の例では学習者の文が正しく書かれていないことを示すため、「OK desu」とだけ書かれている。8、9、10 からは学習者の誤りの傾向が見て取れる。8 では hanasemasu を hanashimasu と書いてしまい、活用の誤りが見られる。また mada を made としており、母音を混同していると推察できる。9 では ha を no と書いており、助詞の選択に困難さがあるようである。10 では americain を americagen と書いており、amerika と america の母音のスパングからくる誤りと、gen の母音の発音からくる誤りが組み合わされている。

Lang-8 から集めたローマ字で書かれたほとんどの学習者の文には、単語と単語の間に区切りが置かれていないという特徴がある。ただしこの接続を除く場合、日本語の文の単語切れ目を判別する。他の特徴として、助詞の発音性がある。例えば、」へ」はローマ字では ha に割り当てられるが、wa と発音されるため、そう書かれることが多い。「へ」を書くときの wo と o、「へ」を書くときの he と e にも同じ発音性がある。

### 5. 誤り訂正ローマ字仮名変換
システムは 3 つの機能からなる。言語の特定、類似語探索による誤り訂正、そしてローマ字仮名変換である。この章では各機能がどのように実現されているかについて詳しく述べる。

---

*1 http://lang-8.com/*
5.1 言語の特性

言語の特性はローマ字で書かれた入力文と、英語の辞書及びローマ字化された日本語辞書との照合によって行われる。"学習者は翻訳によってローマ字化せずに、自分の母語の単語をそのまま書き込むことがある。我々の目的は、完全な音読を行うことではないので、ローマ字化された日本語だけに名前を変換したり、これを実現するため、私たちは英語の単語辞書を用いた。英語の辞書を用いた理由は、学習者の文で見つかる日本語以外の単語は英単語がほとんどだからである。各言語の辞書を用意することで容易に他の言語にも拡張可能である。この辞書にマッチした単語は仮名に変換されないようにタグを付けておく。辞書には 155,287 個の単語を持つ WordNet 2.1 の2版を用いた。誤り訂正しなくても良い正しい単語を除外しておくため、日本語の辞書も用いた。これには IPADic 2.7.0 を使用した。さらに動詞活用の辞書も使用した。なぜなら、学習者の末尾文では動詞その他の活用語尾は結合されているが、我々の日本語辞書内では分割して登録されているためである。この辞書は 1991 年度の每日新聞に登場するすべての動詞の活用形から作成した。日本語の係り受け解析器である CaboCha 0.53×3 を用いて動詞をひとつ以上含む文節を抽出した。これにより抽出された活用形は 243,663 個である。

5.2 誤り訂正

英語の辞書にも日本語の辞書にもマッチしなかった単語を以下的手法によって誤り訂正する。誤り訂正は 2 種類の異なる尺度を用いた類似度探索で行われる。すなわち、文字 uni-gram のコーシャツ類似度と編集距離である。類似度探索用辞書には IPADic のみを用いた。

5.2.1 類似度探索を用いた訂正候補の生成

我々は編集距離が最も小さいものを訂正候補として選択したいが、編集距離とはある文字列を他の文字列に変形するために必要な編集操作の最小の操作数である。ここで、編集操作とは挿入、削除、そして置換のことである。しかし、誤差数が大きい時、編集距離の計算コストが高くなり、システムを動かす上で問題となりうる。そこで、編集距離の計算をする前にコーシャツ類似度による類似度探索で候補数を絞り込むことで計算コストを削減する。

コーシャツ類似度は文字 n-gram の素性を用いて計算されるが、ここで n には 1 を用いた。これにより、辞書内面の適切な候補をほとんどの候補をカバーすることができ、かつ不要な候補数を大幅に削減することが出来るからである。例えば、学習者の学習した単語 packuに対する辞書内の類似度をコーシャツ類似度で探すと、候補の数は 163 個にまで減らすことが出来る。探索された単語の例には kau, pakku, chikau などが含まれる。コーシャツ類似度による探索は大規模な単語辞書に対して非常に効率的に行うことが出来る。

5.2.2 最終候補の選択

文字長によって正規化された文字 n-gram のコストによって最終候補を選択する。この計算はローマ字の文字 5-gram の言語モデルを用いて行う。入力文字列を先頭から 5 文字切り出し、そのコストを言語モデルから求め、末尾までのコストの総和を求めた後、文字長で割ることで入力文脈のコストを求める。言語モデルは 1991 年度の每日新聞を kakasi を用

*1 ローマ字化には kakasi 2.3.4 を用いた。http://kakasi.namazu.org/
*2 http://wordnet.princeton.edu/
*3 http://chasen.org/~taku/software/cabocha/
Table 2: yoroshiku onegai shimasu. yoroshiku onegai shimasu. 77.3% 4 77.3% 4 http://anthy.sourceforge.jp/

6.2 実験設定
結、手の否定、彼女をランダム化した。Anty 7900*2をベースラインとして用いた、これはデファクトスタンダードとなっている日本語入力システムのうちの一つであり、言語特有と類似度探索のどちらも行わない。そこでAntyの名著のために、これは誤り訂正のために特別に作られたものではないことを記しておく。さらに我々の方法に関しても、類似度探索を用いたものと、用いていないものについてそれぞれ実験を行ない、結果について比較した。

6.3 使用データ
3 節で述べたように添削者によって編集の仕方にはばらつきがあるため、Lang-8 上の学習者の文と、添削者の文を単純に対応付けただけでは正しい添削になっている保証がない。そこで、我々は正解データを自分たちで添削し直し作成した。Lang-8 からローマ字数を 500文集めてきて使用した。いくつかの文にはすでに添削者によって修正が加えられていたが、質の均一性を保つため自分たちで添削を付け直した。このとき、学習者の文には様々な誤りが含まれていたが、スペル誤りだけ訂正されたデータを作成した。なぜなら、本論文の主題はスペル誤り訂正を施すことにあるからである。表 2 に作成したデータセットの例を示す。

6.4 実験結果
表 3 にスペル訂正の精度を示す。提案手法の単語精度は 85.0%, Anthy の 74.5% より도 10 ポイント高かった。我々のシステムで類似度探索を用いないかった場合の精度は 84.5% であった。このことより、言語特有は本手法において非常に重要であることがわかり、正しく訂正された単語の例を表 4 に示す。類似度探索を用いないかった場合の再現率は 77.3% であった。
り、類似意探索で誤り訂正することにより 78.6%に改善された。正しく訂正できた単語の例を表4に示す。ここで、下線が引かれた単語は誤りを含むもの、波線が引かれているものは日本語ではない単語を示す。適合率も 76.6%から 78.1%へ、類似意探索を用いることにより向上した。これらの結果から、類似意探索による誤り訂正は再現率を下げることなく適合率を改善することができ、誤り訂正に有効であると言える。しかし、ベースラインでさえ正解率が低く、このタスクは困難であることを示している。

7. 論

表3から読み取れる限り、類似意探索を用いた誤り訂正はローマ字仮名変換の性能を向上させたが、依然この手法には改善の余地が残されている。ここでは誤りの種類を調査し、解決するための案を紹介する。システムが正しく校正できなかった単語のうち、最も多かった3つの誤りのタイプをパターンごとに以下に示す。それは違う単語とのマッチング、入力単語と正しい単語との遠すぎる編集距離、そして複合語である。

### 7.1 異なる単語とのマッチング

入力された文脈が辞書内に存在していたとき、本来学習者が想定していた単語と異なっていてもマッチしてしまう。その例を表7に示す。例えば、学習者が renshuuという単語を書きたかっただけで、誤って renshuuと入力してしまった場合、この単語を修正することはできない。なぜなら、renshuuという文脈は日本語の単語辞書内に存在するからである。この種の問題は、文脈の情報を用いることで改善が期待できるため、単語 n-gram モデルを用いても有効である。

### 7.2 遠すぎる編集距離

入力からの編集距離がある閾値よりも低い単語は訂正候補として選択されない。その事例を表7に示す。たとえば学習者が mizukashiiをmizugashi書いてしまったとするとき、この2つの文脈間の編集距離は3である。これは我々の設定した閾値よりも高い値である。
ため訂正されない。間違えた値をより大きくすることもできるが、大きくずれるとはほどあま
り似ていない単語が候補に増えてきてしまう。この問題は、表5に見られるような学習者の
誤り方の傾向を反映した重み付けを編集距離に与えることで改善されると考えられる。例え
ば、1, 2, 3, 4番の文では doumo を domou と書いてしまう。このような学習者が母音
を混同している例が学習者の文内で多く見られた。そこで母音の置換や入れ替えには低いコ
ストを設定することで、このような音韻的な混同に基づく誤りを訂正できると考えられる。
また、母音の発音の影響と考えられるが、5, 6番の文で見られるように su や shi を書くとき
母音を抜きにして書いているものが多い。7番の文では n の発音を ng と書いている。8, 9
番の文でも母音の発音の影響とみられる誤り方をしている。これらの場合も同様に学習者の
誤り方の傾向を反映した編集距離の重み付けを施すことで精度が改善されるって待てる。例え
ば、子音だけで日本語を入力する手法が文献5) で提案されており、母音の編集距離を低
くしても意図した入力を推測することができる。

7.3 複合語

我々の手法は、辞書の区切りの粒度と学習者の文の区切り粒度が等しい時にのみ有効
である。区切りの粒度が異なるためにうまく訂正できなかった例を表8 に示す。例えば、
nouryokushiken という単語は nouryoku と shiken という2つの単語としても扱うことができ、実際 IPADic で
2つの単語として抜かれており、nouryokushiken という単語を
記録されていない。そのため、マッチングの際に探し当たることができない。この問題を
解決するためには、単語分割の技術を各入力文字列に適用することが有効であると考えて
いる。

8. おわりに

本稿ではローマ字で書かれた学習者の文を仮名に変換する手法を紹介した。我々のシス
テムの目標は言語学習 SNS の添削者が、楽に学習者の文を読めるようにすることである。言
語特定と類似語探索を用いたスペル訂正システムを紹介した。このシステムは学習者のロー
マ字で書かれた文を仮名に変換するタスクにおいて、既存の日本語入力システムよりも 10

ポイント高い精度を達成した。実験結果に対し誤り分析を行なうことで、異なる辞書にマッ
チングしてしまう場合や、編集距離が遠すぎる場合、複合語などにうまく対応できないこと
がわかった。また学習者は母音を混同やすいという傾向や、母音の影響を受けた誤り方を
する傾向があるということがわかった。今度は文脈の情報を利用、文字列を単語分割の手
法で解析し、学習者の誤り傾向を反映する、などして誤り訂正ローマ字仮名変換の精度を高め
たい。

謝辞

今回実験でデータを使用させていただいた Lang-8 を運営されておられる喜洋日々に感
謝いたします。

参考文献

1) Zheng Chen and Kai-Fu Lee. A New Statistical Approach to Chinese Pinyin Input. In Proceedings of ACL, pp. 241–247, 2000.
2) YoEHara and Kumiko Tanaka-Ishii. Multilingual Text Entry using Automatic Language Detection. In Proceedings of IJCNLP, pp. 441–448, 2008.
3) Tomoy 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 IJCNLP, 2011.
4) Naoaki OLAZAKI and Jun’ichi TSUIJI. Simple and Efficient Algorithm for Approximate Dictionary Matching. In Proceedings of COLING, pp. 851–859, 2010.
5) Kumiko Tanaka-Ishii, Yusuke Imutsu, and Masato Takeichi. Japanese input system with digits –Can Japanese be input only with consonants? In Proceedings of HLT, pp. 211–218, 2001.
6) Yabin Zheng, Chen Li, and Maosong Sun. CHIME: An Efficient Error-Tolerant Chinese Pinyin Input Method. In Proceedings of IJCAI, pp. 2551–2556, 2011.
7) 新納浩幸. 仮名形 N-gram による仮名名前の誤り検出とその修正. 情報処理学会論文誌, Vol.40, No.6, pp. 2690–2698, 1999.