視覚的「読み」を用いた分割表記文字の処理

乾 亮　　山村 毅
愛知県立大学 情報科学部

{is151013@cis, yamamura@ist}.aichi-pu.ac.jp

1 はじめに

現代社会では，様々な場所に文章が存在し，一般的に新聞や本などでは，正書法に則った正しい文章が使われている．しかし近年では，LINE や Twitter などの SNS が使われるようになり，そこでは，通常は検閲や校正などは行われないので，様々な正規でない表現が含まれる文章が使われることがある．

自然言語処理は，コンピュータを用いて言語情報を活用する技術を開発することを目的とした研究分野であり，近年の AI ブームにも押されて，様々な領域でその活用が期待されている．しかし，現在の自然言語処理技術は，主として新聞記事などの文章を手本にして技術開発が行われてきたため，インターネット上の文章のように文法的に誤りを含んだ文章にはうまく対処できないという問題がある．このため，誤りに堅牢な自然言語処理技術の開発が強く望まれている．

これまで、文の誤り訂正に関する研究は国内外で広く研究されているが，その多くが言語学習者向けの文法誤りの検出や訂正に関するものである．

例えば，今枝ら [1] および石川ら [2] は品詞の並びに関するルールと格フレームを用いた助詞の誤り検出と訂正の方法を提案している．また，南保ら [3] は，文節内の特徴を助詞と組み合わせてルール化，帰納的学習を用いて抽出されたルール同士から抽象化したルールを新たに自動生成する方法を提案し，これを用いて日本語の助詞誤りの検出・校正を行うシステムを開発している．このほか，大木ら [4] は SVM を用いてシステム開発文書を対象に助詞の誤用を判定する方法を，笠原ら [5] は雑音のある通信路モデルに基づいて助詞の誤り訂正を行う方法を，水本ら [6] はフレーズベースの統計的機械翻訳を用いて日本語学習者の作文誤り訂正を行う方法を，それぞれ提案してる．

一方，これらとは異なり，SNS などにおける崩れた表記を対象とした研究も行われている [7].

例えば，勝木ら [8] は,「ぽっちゃり」などのオノマトペや長音化，小文字表記など，形態素辞書に登録されていない語に対して，既知形態素とのマッチングやパターンに基づいてオンラインで推測する方法を提案している．また，池田ら [9] は，ニューラルネットワークを用いて，崩れ表記を含む日本語文を正規化する方法を提案している．

本研究では，インターネットの文章で見られるような分割表記の文字を含む日本語文章を解析する方法を提案する．

2 分割表記文字

インターネット利用者は多種多様で，中には社会的混乱や特定個人の誹謗中傷のために，正確ではない情報や偽の情報を発信したりする人がいる．特に近年では反社会的な活動を勧誘したりする人もおり，社会問題の一つとなっている．こういった情報を察知することができれば犯罪の防止にもつながり，インターネット利用の安全性も増すであろう．しかし，不法行為を行おうとする人は，監視に引っかからないように文を偽装することがある．その一つに，文字を複数の文字に分割して表記するというものがある．

例えば「死」という文字を「歹」と「匕」に分けて,「おまえ歹匕ねよ」のように表記することで，コンピュータの内部的には「死」という文字を使っていなくても，人間の視覚的には「死」という文字を使っているように見せることができる．これは Twitter などで暴言を書き込むときによく使われる表記である．このように一つの文字を複数の文字に分割して表記した文字列を，本研究では分割表記文字と呼ぶことにする．

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

3 「読み」の概念

我々人間は，未知語や誤りやを含んだ文であっても，その意味を理解できることがある．これは，常識のようなものを用いて意味の解釈を行なっているからだと考えることができるが，同時に，人間が柔軟にその解釈を変更しているからだとも考えることができる．

例えば，一般的な自然言語処理システムでは，入力は文という文字列であるが，文中の各文字に対する解釈は固定的であるため,「王里 由 が 矢口 りたい」における「王」は「おう」である (内部的には Unicode の U+738B)．しかし人間は，これを「王」とも見るし,「理」の一部とも見る．すなわち，文字に対する解釈は固定的ではなく，全体として意味が通じるよう柔軟に解釈する．この人間のもつ柔軟な解釈を本研究では「読み」と呼ぶことにする．

「読み」には，基本的に次の二つが存在すると考えらえる．一つは,「夜露死苦」を「よろしく」と解釈する聴覚的「読み」，もう一つは,「王里」を「理」と解釈する視覚的「読み」である．本研究では後者の視覚的「読み」の概念に基づいて分割表記文字を処理する方法を提案する．

4 提案手法

分割表記文字を含んだ文は，コンピュータの側からは，未知語を含むあるいは誤った文と同じである．このため，一つの方法として，分割表記文字に関する特別な辞書を用意し，これを用いて解析するというものが考えられる．この方法は，いわゆるルールベースの方法と位置付けられ，ルールを充実すれば高い精度で解析を行うことができるという利点があるが，反面，ルールの保守・更新に手間がかかるという問題点がある．

そこで，3 で導入した視覚的「読み」の概念を利用することを考える．

人間は，分割表記されている場合でも，その見た目から元の (分割前の) 文字を推定していると考えることができるため，記号列で表された文を目で見た時のイメージである画像に変換し，その変換された画像に対して文字認識処理を行えば，分割表記文字を処理することができるのではないかと考えられる．しかし，分割表記文字は通常の大きさの文字を複数組み合わせたものであるから，文をそのまま画像に変換し文字認識しただけでは，別々の文字として認識されてしまう (「王里由」は「王」,「里」,「由」とバラバラに認識されてしまう)．そこで，これに対処するため，分割表記文字が用いられている部分を特定し，その部分についてのみ画像変換・文字認識をするようにする．先にも述べたように，分割表記文字の部分は，コンピュータの側からは,「誤り」と同じであるので，これまでの自然言語処理システムで用いられてきた誤り検出の方法を用いれば，特定することができる (例えば，林ら [10] は，bi-gram 辞書を利用して漢字変換誤り検出を行う方法を提案している)．

本研究では，文字の bi-gram を利用して，文中から分割表記文字部分を抽出し，これを画像に変換したあと，文字認識処理を行なって，分割表記文字を元の文字に変換する．

具体的な処理手順は以下の通りである．

1. 文字の bi-gram 辞書をコーパスから計算して用意しておく
2. 入力文の bi-gram を先頭から順に取り出して並べる
3. bi-gram 辞書を用いて入力文の bi-gram で低頻度または存在しないもの (ゼロ bi-gram) を検出する
4. 検出されたゼロ bi-gram の位置から分割表記文字候補の 2 文字を取り出す
5. 取り出した文字をそれぞれ画像にし，サイズや間隔を調整して一つの画像にする
6. 画像を文字認識し，入力文の対応する位置に戻す

手順 4 において，分割表記文字の 2 文字の取り出し方は以下の通りである．

- 検出されたゼロ bi-gram が 1 つの場合は，それに対応する位置の 2 文字を分割表記文字候補として取り出す.
- 検出されたゼロ bi-gram が 2 つの場合は，まずはどちらか一方について手順 5，6 を行ない，生成された文に手順 2，3 と同様の処理を行ってゼロ bi-gram が検出されなければ，それをそのまま最終結果とする．ゼロ bi-gram が検出されれば，残りの一方について同様の処理を行う．いずれにおいてもゼロ bi-gram が検出されてしまった場合は，分割表記文字の取り出しをしない.

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

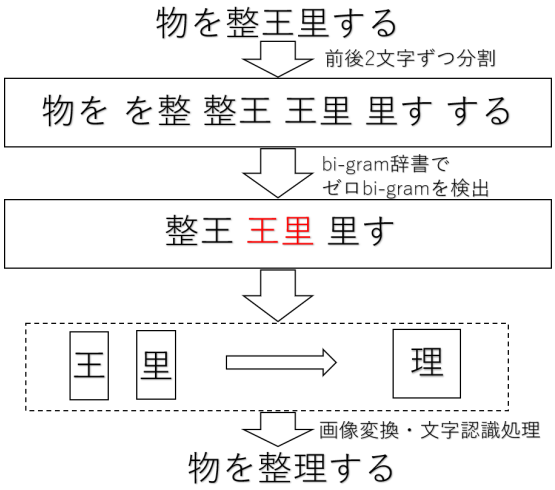

図 1: システムの流れ

- 検出されたゼロ bi-gram が 3 つの場合は，それらの中間に位置するゼロ bi-gram に対応する位置の 2 文字を分割表記文字候補として取り出す．

5 評価実験

5.1 システム概要

4 で述べた方法を Python を用いて実装した．このシステムにおいて，文字を画像に変換したり，調整したりするのには，Python の PIL ライブラリを利用した．また，文字の画像認識には Python の pyocr ライブラリを使用した．分割表記文字の検出を行うために必要な bi-gram 辞書の作成には，毎日新聞の 2009 年，2010 年の 2 年分の記事を用いた．

システムの処理の流れを図 1 に示す．この図の例では bi-gram 辞書で低頻度または存在しないもの (ゼロ bi-gram) を検出した結果,「整王」,「王里」,「里す」の 3 つの並んだ部分が検出されている．次にこれらの結果から,「整王」と「里す」で挟まれた「王里」を分割表記文字として取り出し，これを画像に変換，文字認識を行う (「理」となる)．最終的には，入力文は「物を整理する」となった．

5.2 実験方法

Twitter の文を抜粋して作った 100 文を対象に，5.1 で作成したシステムで評価実験を行った．なお，本研究では 1 文につき分割表記文字が 1 つの文を対象としている．

表 1: 評価実験結果

成功した文	失敗した文	合計
69	31	100

表 2: 失敗の内訳

内訳	文数
文字認識の失敗	19
ゼロ bi-gram 未検出	5
ゼロ bi-gram の過剰検出	7

5.3 実験結果

結果を表 1 に示す．
以下に成功した例を次に示す．

早く歹ヒね → 早く死ね
糸エ白に出る → 紅白に出る
無王里怖い → 無理怖い

5.4 考察

表 1 の失敗した 31 例について，その内訳を調べたものを表 2 に示す．以下これらについて考察する．

文字認識の失敗

これは，分割表記文字部分の検出，画像変換はできているが，文字認識の部分で想定していない結果が出力されたことによる失敗である．

早くタ匕ね → 早く叱ね
無理小布い → 無理小右い
横シ兵が楽しい → 横シ兵が楽しい

このような例に対しては，文字認識で複数候補を出し，それらの候補を用いて生成した文の妥当性を (例えば 4 で述べた bi-gram 辞書によるものと同様の方法で) 調べることで対応できるのではと考えている．

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

ゼロ bi-gram 未検出

これは，4 で述べた手順 3 においてゼロ bi-gram が見つからなかったことによる失敗である．

私は魚が女子き → 私は魚が女子き

参カロさせてください → 参カロさせてください

日月日は晴れ → 日月日は晴れ

「好」を「女子」,「加」を「カロ」,「明」を「日月」とそれぞれ分割表記しているが，これらが bi-gram 辞書に高頻度で存在していたため検出されなかった．これについては，bi-gram 辞書ではなく tri-gram 辞書を使用して検出するなどの方法が考えられる．

ゼロ bi-gram の過剰検出

これは，4 で述べた手順 3 において，分割表記文字でない通常の文字列が誤ってゼロ bi-gram として検出されてしまったことによる失敗である．

言川練する → 言幟する

イ為サイト → イ為サイト

野球選手に小童れる → 野球選手に小童れる

例えば,「野球選手に小童れる」の場合,「小童」,「童れ」の部分をゼロ bi-gram として検出し，(ゼロ bi-gram が 2 つだったので),「小童」を文字認識して「憧」としたのち，元の文に当てはめて，再度ゼロ bi-gram の検出を行うが (「野球選手に憧れる」となるが),「に憧」,「憧れ」が，作成した bi-gram 辞書に高頻度では存在しなかったため失敗してしまった (その後「童れ」を文字認識し同様に失敗).

これについては，辞書を作成するときの新聞記事を増やすなど，データを増やせば対応出来るのではないかと考えている．

6 おわりに

本研究では，視覚的「読み」の概念を導入し，文字認識を取り入れることで分割表記文字を含む日本語文章を解析する方法を提案した．文字認識を使うことは有効ではあったが，検出の仕方，文字認識結果の複数候補の出力など課題があることがわかった．

今後は，本研究で得られた課題に取り組み，また，1 文につき分割表記文字が 2 つ以上の文の場合についても研究を取り組みたい．

参考文献

[1] 今枝 恒治, 河合 敦夫, 石川 裕司, 永田 亮, 桝井 文人: “日本語学習者の作文における格助詞の誤り検出と訂正”, 情報処理学会研究報告, CE68-6, pp.39–46, 2003

[2] 石川 裕司, 河合 敦夫, 多田 直人, 永田 亮, 桝井 文人: “日本語学習者の作文における格助詞の誤り検出と訂正”, 情報処理学会第 66 回全国大会講演論文集, No.2, pp.323–324, 2004

[3] 南保 亮太, 乙武 北斗, 荒木 健治: “文節内の特徴を用いた日本語助詞誤りの自動検出”, 情報処理学会研究報告, Vol.2007-NL-181, No.17, pp.107–112, 2007

[4] 大木 環美, 大山 浩美, 北内 啓, 末永 高志, 松本裕治:“非日本語母国話者の作成するシステム開発文書を対象とした助詞の誤用判定”, 言語処理学会第 17 回年次大会発表論文集, pp.1047–1050, 2011

[5] 笠原 誠司, 藤野 拓也, 小町 守, 永田 昌明, 松本 裕治: “日本語学習者の誤り傾向を反映した格助詞訂正”, 言語処理学会第 18 回年次大会発表論文集, pp.14–17,2012

[6] 水本 智也, 小町 守, 永田 昌明, 松本 裕治: “日本語学習者の作文自動誤り訂正のための語学学習 SNS の添削ログからの知識獲得”, 人工知能学会論文誌, Vol.28, No.5, pp.420–432, 2013

[7] 笹野 遼平, 鍛治 伸裕: “新しい語・崩れた表記の処理”, 情報処理, Vol.53, No.3, pp.211–216, 2012

[8] 勝木 健太, 笹野 遼平, 河原 大輔, 黒橋 禎夫: “Web 上の多彩な言語表現バリエーションに対応した頑健な形態素解析”, 言語処理学会第 17 回年次大会発表論文集, pp.1003–1006, 2011

[9] 池田 大志, 進藤 裕之, 松本 裕治: “Encoder-Decoder モデルを用いた日本語崩れ表記の正規化”, Vol.2016-NL-228, No.11, pp.1–6, 2016

[10] 林 秀治, 山本 和英: “漏れのない漢字変換誤り検出と誤り可能性によるレベル分け”, 言語処理学会第 22 回年次大会発表論文集, pp.1145–1148, 2016

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.