マイクロブログを対象とした形態素解析誤りの自動検出と誤り分析

宮里 貴之　　白井 清昭
北陸先端科学技術大学院大学 先端科学技術研究科
{t.miyazato,kshirai}@jaist.ac.jp

1 はじめに

近年，Twitter[1] を始めとするマイクロブログは，評判情報分析やテキストマイニングのための情報源として注目を集めている．多くの自然言語処理と同様に，マイクロブログのテキストから情報を取り出す際に行われる最も基本的な処理は形態素解析である．しかしながら，既存の形態素解析ツールの多くは新聞記事のような書き言葉のテキストを対象として開発されているため，これをウェブ上のテキストやマイクロブログのテキストに適用したときに形態素解析の正解率が低下することが知られている．

大崎らは，Social Network Service(SNS) 上のテキストを対象とした形態素解析のためのコーパスとして，Twitter に投稿されたテキストに形態素情報を付与したコーパスを構築した [5]．また，このコーパスを構築する際に，形態素解析ツール Kytea[4] を用いて Twitter のテキストを解析したときに生じる誤りを分析し，その分析結果を基にタグ付けの指針を決定している．Twitter を解析する際によく生じる誤りとして，表記ゆれ，長音を含む語，造語，接頭辞・接尾辞などに起因する誤りを報告している．

ウェブ上のテキストの形態素解析の性能を向上させる研究も行われている．勝木らは，形態素解析の誤りを生じやすい未知語を分類し，未知語を正しく処理するための方法を論じた [2]．具体的には，非反復型オノマトペ，長音化表記の未知語，小文字表示の未知語を対象に，正しい形態素解析結果を得るための手法を提案した．池田らは，ウェブテキストにおける未知語に対して，その前後に出現する文字列をくだけた表現の少ない文書集合から検索し，未知語に対する形態素解析結果を修正するルールを自動生成した [1]．

マイクロブログのテキストに対する形態素解析の性能を改善するためには，まず既存手法によって生じる解析の誤りを分析し，その分析結果を基に形態素解析のアルゴリズムを改良するという手続きが一般的である．しかし，マイクロブログには多種多様な表現が出現することから，誤りの種類も多岐にわたると考えられる．形態素解析の誤りを分析した先行研究 [2, 5] もあるが，マイクロブログを対象としたより頑健な形態素解析ツールを開発するためには，大規模なコーパスを対象とした解析誤りの分析が必要であろう．

本論文では，人手による形態素解析誤りの分析の負荷を軽減するために，Twitter に投稿されたテキスト (ツイート) における形態素解析の誤りを自動的に検出する手法について述べる．様々な形態素解析ツールの誤り分析に適用できる汎用的な手法を提案する．さらに，提案手法を適用して検出された誤り箇所を人手でチェックし，形態素解析の誤り分析を行う [3]．

2 形態素解析誤り箇所の自動検出

本節では，ツイートを形態素解析ツールで解析したときに生じる誤りの箇所を自動的に検出する手法について述べる．ただし，検出対象とする誤りを以下に限定する．

- 単語分割の誤り箇所を検出する
  形態素解析の誤りには，大きく分けて，単語分割の誤りと品詞付与の誤りがある．本論文では，品詞付与の誤りの自動検出は今後の課題とし，単語分割の誤りのみを対象とする．
- ひらがな列の単語分割の誤りを検出する
  予備調査の結果，ツイートの形態素解析の誤りは，ひらがな列を正しく分割できないケースがほとんどであることがわかった．そのため，本論文ではひらがな列の単語分割の誤りのみを対象とする．

単語分割の誤りを検出する手続きは以下の通りである．

1. 形態素解析
   既存の形態素解析ツールを用いてツイートの形態素解析を行う．以下，得られた単語列を $w_1 \cdots w_n$

[1] http://twitter.com/

と表記する．また，単語 w_i と w_{i+1} の境界を b_i $(1 \leq i \leq n-1)$ と表記する．

2. 誤り箇所の候補の検出
   前後の単語 w_i と w_{i+1} がともにひらがな列であるような全ての単語境界 b_i を誤り箇所の候補とする．
3. 誤りの判定
   b_i が以下の条件を同時に満たすとき，それを単語分割の誤り箇所として検出する．

$$O_{news}(w_i, w_{i+1}) \leq T_{news} \quad (1)$$

$$O_{tw}(w_i \cdot w_{i+1}) \leq T_{tw} \quad (2)$$

式 (1) における $O_{news}(w_a, w_b)$ は，新聞記事コーパスにおける単語 bi-gram の出現頻度を表わす．新聞記事コーパスをあらかじめ形態素解析し，ひらがなのみで構成される単語が連続して出現するとき，その単語列の頻度を O_{news} として記録しておく．もし，単語境界の前後に出現する単語列が，新聞記事の中であまり出現しないとき，その単語境界は誤りである可能性が高いと考えられる．そのため，単語 bi-gram の出現頻度が閾値 T_{news} よりも小さいとき，その単語境界を誤りとみなす．

新聞と Twitter では話題や書き方のスタイルが異なることに注意する必要がある．新聞記事ではあまり取り上げられない話題に言及したツイートや，Twitter 特有の表現が用いられているときには，形態素解析の結果は正しいのにも関わらず，新聞記事の出現頻度が低くなる可能性がある．したがって，式 (1) の条件だけを考慮すると，正しい単語境界をエラーと誤判定しやすい．そのため，式 (2) の条件を加える．

式 (2) における $O_{tw}(s)$ は，ツイートコーパスにおける文字列 s の出現頻度を表わす．また，$w_i \cdot w_{i+1}$ は，w_i と w_{i+1} を連結した文字列を表わす．あらかじめ大量のツイートをダウンロードし，そのツイートコーパスにおける文字列 $w_i \cdot w_{i+1}$ の出現頻度が閾値 T_{tw} よりも低いとき，単語境界を誤りと判定する．新聞記事では単語 bi-gram の出現頻度で誤りの有無を判定しているが，ツイートコーパスでは文字列の出現頻度で判定している．これは，ツイートでは新聞記事と比べて形態素解析の誤りが多いと考えられるため，ツイートコーパスから得られる単語 bi-gram の出現頻度の信頼度が低下することを考慮したためである．実装では，ツイートコーパスに対して suffix array を構築することで，任意の文字列の出現頻度を高速に調べる．

3 誤り検出手法の評価

本節では提案手法の評価実験について述べる．単語分割の誤り箇所を検出するために以下の 3 つのコーパスを用意した．

- 新聞記事コーパス
  毎日新聞の 1991 年から 2010 年の 20 年分の新聞記事データ．式 (1) における O_{news} を得るため，形態素解析ツール JUMAN[2] を用いて形態素解析し，ひらがなのみで構成される単語の連続を単語 bi-gram として抽出し，その一覧を出現頻度とともに保存する．単語 bi-gram ののべ数は 86,842,897，異なり数は 449,956 であった．
- ツイートコーパス
  Twitter に投稿されたツイートを収集したコーパス．まず，上述の新聞記事コーパスにおける出現頻度上位 100 件の名詞をクエリとし，Twitter API を用いてそのクエリを含むツイートを取得する．ひとつのクエリにつき取得するツイート数は最大 18,000 件とした．重複を除いて 528,486 件のツイートを収集した．このコーパスは式 (2) における O_{tw} を得るために用いる．
- 評価用コーパス
  形態素解析の誤り箇所を人手で付与した評価用データ．まず，Twitter API を用いて，「学校」もしくは「勉強」を含むツイートを 342 件収集した．これらのツイートを JUMAN で形態素解析し，その誤り箇所を人手で付与した．単語分割の誤り箇所は 53 件，そのうちひらがな列の分割の誤り箇所は 51 件であった．

閾値 T_{news} と T_{tw} を 1 から 20 まで 1 刻みで変動させたときの精度と再現率の変化を図 1 に示す．T_{news} と T_{tw} は異なる値を設定できるが，この実験では両者に同じ値を設定している．一般に，閾値を大きく設定すると，誤り検出箇所が増え，精度は低下するが再現率は向上する．図 1 のグラフを見ると，精度は，閾値が 4 を越えると単調に減少するが，閾値が 1,2,3,4 のときは 0.083, 0.080, 0.087, 0.10 となり，増減している[3]．一方，再現率は，閾値を大きくすると単調に増加するが，4 を越えると 0.384 から変化しない．なお，閾値を 30 以上に設定すると再現率が増加することを確認した．

[2] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

[3] 誤り検出箇所もその中で正しく誤りを検出できた数も (精度の分母も分子も) 単調に増加していることは確認した．

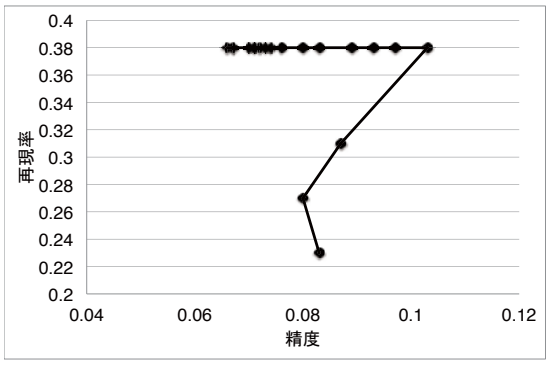

図 1: 誤り箇所検出の精度と再現率

$T_{news} = T_{tw} = 4$ と設定したとき，単語分割誤り検出の精度は 0.10，再現率は 0.38，F 値は 0.16 であり，$T_{news} = T_{tw} = 10$ のときの精度は 0.076，再現率は 0.38，F 値は 0.13 であった．本論文では，汎用的かつ基本的な手法として，単語 bi-gram や文字列の頻度に基づく手法を提案したため，精度や再現率は十分に高いとは言えない．

4 単語分割の誤り分析

本節では提案手法を用いた単語分割の誤り分析について報告する．まず，528,486 件のツイートコーパスに対して提案手法 ($T_{news} = 10$，$T_{tw} = 10$ と設定) を適用したところ，51,540 件のツイートに対し単語分割の誤りが検出された．この中から 300 件のツイートをランダムに選択し，自動検出された誤り箇所を人手でチェックし，単語分割が実際に誤っているか否か，誤っている場合にはその誤りの要因を分析した．

表 1 は誤りの要因とそれに該当するツイート数を示している．以下，それぞれの誤り要因の詳細と誤り例を紹介する．

表 1: 提案手法を用いた単語分割の誤り分析

口語表現	22
ひらがな表記された固有名詞	13
方言	4
句点の省略	3
誤字	3
語尾の変形	3
ひらがなの連続による強調	2
文の途中に出現する動詞の命令形	1
ネットスラング	1
品詞の誤り	3
正解が不明	4
合計	59

口語表現

口語表現やくだけた表現に対する単語分割の誤りが多かった．以下のツイート T1 では,「わけない/じゃん」という区切りが正しいが,「わけな/い/じゃ/ん」と誤って解析されている[4].

T1 腹 (名詞) 弛ま (動詞) ない (接尾辞) わけな (助動詞) い (動詞) ⊗ じゃ(助詞) ん (未定義語)

ひらがな表記された固有名詞

固有名詞がひらがなで書かれているとき，その固有名詞のひらがな表記が形態素解析の辞書に登録されていないために単語分割の誤りが生じていた．通常は漢字で表記される固有名詞がひらがなで表記されている例も含む．例を以下に挙げる．

T2 この (指示詞) ⊗ は (名詞) ちゃん (接尾辞) の (助詞) 仕事 (名詞) の (助詞) 姿勢 (名詞) は (助詞) 見習わ (動詞) ねば (助動詞)

方言

方言も辞書に登録されてないために誤りが生じやすい．特に関西弁の誤りが多かった．例を以下に示す．

T3 RT(未定義語) し (動詞) ます (接尾辞) い (動詞) ⊗ うて (動詞) ⊗ も (助詞)

句点の省略

句点が省略され，前の文の文末と次の文の文頭がつながったとき，そこで単語分割の誤りが生じる場合があった．T4 の例では,「怒りだす」の後の句点が省略されている.「だから」は文頭に出現しやすい接続詞であるが，句点が省略されているために文の途中と認識され，単語分割に失敗している．

T4 ... と (助詞) 感じた (動詞) とき (名詞) に (助詞) 人 (名詞) は (助詞) 怒り (動詞) だす (動詞) だ (名詞) ⊗ から (助詞) 【(特殊) わかって (動詞) 上げる (動詞) こと (名詞) 】(特殊) ...

誤字

誤字によって単語分割の誤りが生じることがある．T5 では,「投稿していただく」を「投稿てただく」と書き誤っているために単語を正しく分割できていない．

T5 投稿 (名詞) て (名詞) ⊗ ただ (副詞) ⊗ く (名詞) と (助詞) ありがたい (形容詞) です (助動詞)

語尾の変形

小文字に直したり長音を加えたりするなど，用言の語尾が変形されているときに単語分割の誤りが生じた．以下に例を挙げる．

T6 姿勢 (名詞) に (助詞) よって (動詞) は (助詞) 腰 (名詞) に (助詞) くる (動詞) じ (名詞) ⊗ ぇ(未定義語)

[4]以下，ツイートの例を載せるときは，括弧で各単語の品詞を，⊗ で自動検出された誤り箇所を示す．

ひらがなの連続による強調

ひらがなを重ねた強調表現が用いられるときに単語分割の誤りが生じた．以下に例を挙げる．

T7 ああ (指示詞) ⊗ あ (感動詞) ありがとう (感動詞) ございます (接尾辞)

文の途中に出現する動詞の命令形

T8 における「逃げろみんな」は倒置表現だが，これにより命令形の動詞が文中に出現する．通常，命令形の動詞は文末に現われるために解析誤りが生じていると考えられる．また，提案手法では検出できなかったが，T8 の冒頭の「逃げろ連呼」のところでも同様の解析誤りが生じている．

T8 #nhk(未定義語) の (助詞) 逃げ (動詞) ろ (名詞) 連呼 (名詞) は (助詞) 震災後の災害に対する姿勢の変化として評価できる と (助詞) 思う (動詞) 逃げ (動詞) ろ (名詞) ⊗ みんな (副詞)

ネットスラング

ネットスラングの単語分割を誤った例が 1 件見つかった．以下の T9 では「ふぁぼした」の単語分割に失敗している．

T9 ふ (名詞) ぁ(未定義語) ぼ (名詞) ⊗ した (動詞) 人 (名詞) に (助詞) 一言 (名詞) 物 (名詞) 申す (動詞)

品詞の誤り

単語分割は合っているが，品詞が正しく同定できなかった事例である．提案手法は単語分割の誤りの検出を目的としおり，このようなエラーは検出の対象とはしていないが，参考として報告する．

正解が不明

どのような単語分割が正しいか判断がつかない事例が 4 件あった．例を以下に示す．

T10 (○˚ ▽˚) 宇宙戦で MS が姿勢制御のためノズルからプシュプシュ出してるのも 紫 (名詞) 汁 (名詞) だ (名詞) うら (名詞) ⊗ っ(未定義語) きー (接尾辞) ♪ (特殊)

人手による誤り分析では，300 件のうち 59 件のツイートから形態素解析の誤りが見つかった．したがって誤りの自動検出の精度はおよそ 0.20 であり，3 節で報告した精度 (0.076) よりも高い．また，誤り分析の際にはツイートの全文を読む必要はなく，自動検出された誤り箇所だけをチェックすればよい．本論文で述べた単語分割の誤りを自動検出する手法は，ツイートを対象とした形態素解析の誤り分析の労力を軽減することに貢献する．

本節で報告した誤り要因のうち，「句点の省略」「ひらがなの連続による強調」「文の途中に出現する動詞の命令形」は，文献 [2] や [5] では報告されていない．1 節で述べたように，マイクロブログには多種多様な表現が出現することから，まだ知られてない誤りの要因も存在すると考えられる．今回の 300 件のツイートを対象とした小規模な誤り分析でも新しい誤り要因が発見できたことから，マイクロブログを対象とした形態素解析の誤り分析は十分に行われているとは言い難く，今後も継続して取り組むべき課題と言える．

5 おわりに

本論文では，ツイートを形態素解析したとき，ひらがな列で発生する単語分割の誤りを自動検出する汎用的な手法を提案した．また，人手による単語分割の誤り分析を実施し，新しいタイプの誤りの要因を発見した．今後の課題として，より大規模なツイートコーパスに対して誤り分析を行い，できるだけ多くの誤り要因を発見すること，その分析結果をもとに形態素解析ツールの性能を向上させる手法を考案することが挙げられる．また，本論文では単語分割の誤りのみを対象としたが，品詞付与の誤りについても，それを自動検出する手法の開発や，自動検出手法を利用した人手による大規模な誤り分析が必要である．

参考文献

[1] 池田和史, 柳原正, 松本一則, 滝嶋康弘. くだけた表現を修正するための教師なし学習方式の提案と評価. 情報科学技術フォーラム講演論文集, Vol. 8, No. 2, pp. 13–18, 2009.

[2] 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.

[3] 宮里貴之. マイクロブログを対象とした形態素解析誤りの自動検出と誤り分析. 修士論文, 北陸先端科学技術大学院大学, 2017.

[4] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 529–533, 2011.

[5] 大崎彩葉, 唐口翔平, 大迫拓矢, 佐々木俊哉, 北川善彬, 堺澤勇也, 小町守. Twitter 日本語形態素解析のためのコーパス構築. 言語処理学会第 22 回年次大会発表論文集, pp. 16–19, 2016.