1F4-3

Twitter 上の「おはよう」を例とした崩れた表記の検出と分析

Detecting and Analyzing Variations of "Good Morning" Phrases on Twitter

藤沼祥成[*1] 横野光[*2] 相澤彰子[*1*2]
Yoshinari Fujinuma Hikaru Yokono Akiko Aizawa

[*1]東京大学 The University of Tokyo
[*2]国立情報学研究所 National Institute of Informatics

Recently, Twitter has attracted a lot of users all over the world. However, the performance of Japanese tokenizers decreases on Twitter because of frequently appearing informal words. We assume that such informal words are created based on some transformation rules caused by sentiments of a person. Our final goal of this research is to extract transformation rules and attach sentiments to it. As a first step towards the final goal, this paper proposes a CRF-based method to extract informal variations of "Good Morning" phrases in Japanese and extract rules used to transform a formal form of "Good Morning" phrase to an informal form.

1. はじめに

Twitter に代表されるソーシャルネットワークには、新聞等の公式な文書では用いられない「崩れた」表記が多く存在する。特に日本語では、ひらがな、カタカナ、漢字、ローマ字と文字の種類が多く、崩れた表記についても、多数のバリエーションが使われている (表 1)。

ここで、多くの日本語解析ツールは入力として標準的なテキストを想定しているため、崩れた表記を含むテキストをそのままの形で与えると、解析の精度が落ちてしまうという問題がある。このため従来から、あらかじめ崩れた表記 (「おはにょ〜」) を辞書に登録されている標準的な表記 (「おはよう」) に変換した上で、形態素解析や構文解析を適用する方法 [Han 11, Liu 11] や、解析ツールに正規化ルールを組み込む方法 [笹野 12] などが用いられている。

しかしながら、このような正規化を適用してしまうと、もともとの崩れた表記に含まれていた発信者の感情などの情報が失われてしまう。Matsumoto ら [Matsumoto 11] は若者言葉には感情が含まれていることを示した。本稿では、同様に崩れた表記にも感情が含まれていることを前提として、崩し方ルールと感情表現の対応付けの可能性を検討する。具体的には Twitter 上に頻出している単語である「おはよう」に注目し、まず、Conditional Random Fields(CRF) モデルを使用して「おはよう」の崩れた異表記を抽出し、次に抽出結果に基づき崩れのルールを抽出することを試みる。

以下本稿では、2 節で関連研究をあげ、3 節で提案手法の概要をまとめる。次に、4 節で実験と評価、5 節で考察を示し、6 節でまとめと今後の課題を述べる。

2. 先行研究

ツイートに対して極性を抽出する研究は盛んに行われている。英語では Brody ら [Brody 11] が文字の繰り返しが極性の強さを表すことを示した。

日本語においては水岡ら [水岡 11] は時間的近傍性を利用し、テレビ番組に対する感想を述べたツイートより感情の言い換え表現を広く収集した。またツイートの文体、すなわち口語調、文語調、男性的、女性的等の特徴に注目し、前田ら [前田 12] はツイートの文体に対応してどのような感情が含まれているかを実験で示した。

本研究の目的は前田らと同じくツイートより極性を抽出することであるが、文体ではなく、一つの単語がどのように変化しているかに注目して検討を行った。

表 1: #ohayo より抽出した「おはよう」の崩れた異表記例

おはヨーグルト	ォ’‘ョ
おはヨン様	おはよう 5 ざいます
はおよーございまーーーーーーーす	おはおはおッはよーん
おはにょ〜	オッハヨ〜ございます
おは養老乃瀧	お廃炉でございます
はようございまーちゃん	おっはろー

3. 提案手法

本稿で提案する手法の目的は 2 点ある。1 点は「おはよう」に相当する崩れた異表記を与えられたツイート中から抽出する事である。もう 1 点は崩れた異表記から崩れルールを抽出する事である。本節で提案する手法の概要は以下の通りである。

1. Twitter データに対して前処理を行う。
2. 訓練した CRF モデルを用いてツイートより「おはよう」の崩れた異表記を抽出する。
3. 抽出した崩れた異表記より、標準的な表記から崩れた表記への崩れルールを抽出する。

3.1 Twitter データに対する前処理

本研究では自動的に生成されたツイートに感情を対応づけることは適当ではないと考え、前処理として手順 1 では [藤沼 12] と同じ手法でそのようなツイートの除去を行った。Twitter にはボットと呼ばれる自動的にツイートするプログラムがあり、ボットからのツイートを削除するため、ボットの性質を利用し

連絡先: 藤沼祥成，東京大学，東京都文京区本郷 7-3-1，y_fujinuma@nii.ac.jp

表 2: 「オハヨゥー♪」に対する素性とラベル (n=2 の場合)

入力文字	素性	タグ
オ	お、おは、オ、オハ	B
ハ	おは、はよ、オハ、ハヨ	I
ヨ	はよ、よう、ハヨ、ヨゥ	I
ゥ	よう、うー、ヨゥ、ゥー	I
ー	うー、ー♪、 ゥー、ー♪	I
♪	ー♪、♪	I

た。ボットはある特定のクライアント*1のみからツイートするため、上位 43 クライアント以外のツイートは取り除いた。また、Twitter 固有の文字列である、URL、ハッシュタグ、@に続くユーザー名に関しては削除し、改行は空白に置き換え、非公式リツイートを取り除くため、RT、Rt、QT 以後の文字列は削除した。これらの処理を行った後、重複しているツイートも削除した。

3.2 CRF を用いた「おはよう」の崩れた異表記抽出

手順 2 では文字に対する系列ラベリングタスクを解決するモデルとして、Conditional Random Fields(CRF) を用いた。ラベルとしては IOB2 タグ、すなわち B タグ、I タグ、O タグを使用した。「おはよう」の崩れた異表記の初めの 1 文字に対して、B タグを振り、B タグ以降の崩れた異表記中の文字には I タグ、それ以外の文字には O タグを振った。他に使用するモデルの選択肢として、挙がるのが Hidden Markov Model(HMM) である。しかし、HMM は生成モデルであるため、互いに独立でない素性を扱う事ができない。本稿では素性として互いに独立でない複数の部分文字列を用いるため、CRF を採用した。

CRF において用いた素性は長さ n の部分文字列と、その部分文字列に対してカタカナと「ぁ」等の小書き文字をひらがなに変換した文字列である。これらの素性を使用した理由はカタカナや小書き文字が用いられた崩れた異表記に対して精度を上げるためである。

使用した部分文字列について $n = 2$ の場合を例に挙げ、述べる。表 2 に「オハヨゥー♪」に対し、素性として長さ 2 の部分文字列を使用した例を示した。今、観測文字列に対し、タグを振る文字を x_0 とする。x_0 のタグを振るに当たり、直前および直後の文字である x_{-1} と x_1 を用いて、長さ 2 の部分文字列 $x_{-1}x_0$ と x_0x_1 を素性として用いた。長さ 3 の部分文字列を使用する時は部分文字列 $x_{-2}x_{-1}x_0$、$x_{-1}x_0x_1$ そして $x_0x_1x_2$ を使用した。

また、モデルがラベルを振る際に見る前後 n 文字に対し、最適な n の値は必ずしも自明ではない。なぜならば n の値を大きくした方が、より多くの種類の崩れた異表記を抽出できそうだが、一方で素性が疎になることにより性能が上がらないからである。そこで本稿では、「おはよう」のスパンの認識や崩れルールの抽出における n の値の影響を後出の実験により確認する。

3.3 崩れルールの抽出

手順 3 の崩れルールとは挿入、削除、置換の 3 つから成るものとする。また、崩れルールは文字単位で適用されるものであり、連続した文字すなわち部分文字列にも適用できるものとする。本稿では崩れルールは “変換前の文字列”→ “変換後の文字列” という方式で示す。例を挙げると、“お”→ “” という崩れルールを「おはよう」に適用した場合、「お」を「」に置換し、「はよう」という文字列に変換する。

崩れルールの抽出手法は編集距離に基づいた手法である。最初に一番最後の文字同士を比較していき、比較している文字同士が異なっている場合、挿入、削除、置換いずれかの崩れルールを生成する。崩れルールを生成した後、連続した、挿入、削除、置換の崩れルールがあった場合には 1 つの崩れルールとしてまとめる。

「おはよう」という文字列から「ハヨウ」という文字列の崩れルールを抽出する例を挙げる。最初に両方の最後の文字である「う」と「ウ」に対し、“う”→ “ウ” という置換する崩れルールを抽出する。同様に “よ”→ “ヨ” と “は”→ “ハ” を抽出する。次に、「ハヨウ」という文字列は長さ 3 であるため、長さ 4 の「おはよう」に対して「お」に対応する文字がない。その結果、“お”→ “” という削除する崩れルールを抽出する。最後に “は” → “ハ”、“よ”→ “ヨ”、そして “う”→ “ウ” は同じ、置換する崩れルールであるため、まとめて、“はよう”→ “ハヨウ”、という崩れルールにする。この例より抽出できた崩れルールは “はよう”→ “ハヨウ” と “お”→ “” である。

4. 実験

前章で述べた「おはよう」の崩れた異表記を抽出する提案手法の性能を検証した。手法の実装には CRFsuite[Okazaki 07] を用いた。また、得られた崩れた異表記がどのような崩れルールに基づいて変換されたかを抽出する実験を行った。

CRF の学習データとして、2012 年 2 月 10 日午前 8 時よりランダムサンプリングした 300 ツイートを人手でアノテーションしたデータを用いた。また、性能を検証するテストデータとして、2 月 18 日午前 8 時よりランダムサンプリングした 10000 ツイートを用いた。

CRF の素性として用いる部分文字列の長さ n の影響を調べるため、実験ではベースラインと $n = 1, 2, 3$ の 3 通りの場合で検証した。ベースラインとしてはツイート中の全ての部分文字列に対し、「おはよう」との編集距離が 1 になる最長の部分文字列を抽出する手法を採用した。その後、顔文字は「おはよう」の一部では無いため、最後の 1 文字が顔文字である場合は除いた。

4.1 評価指標

評価指標としては崩れた異表記の種類を考慮した Precision と、抽出できた崩れた異表記の種類を用いた。異表記の種類を評価指標として用いた理由は、より多様な崩れルールを抽出するのが重要だからである。また、Recall の代わりの指標として抽出した崩れた異表記の種類を用いた理由は 10000 ツイートに対して「おはよう」の崩れた異表記を抽出したため、Recall の算出にアノテーションの人手コストがかかるからである。Recall は

$$\text{Recall} = \frac{\text{正しく抽出した崩れた異表記の種類}}{\text{テストデータ中に存在する崩れた異表記の種類}}$$

であり、本稿では多くの種類の崩れた異表記を抽出する事が目的の 1 つであるため、Recall の代わりの指標として採用した。

また、Precision を算出するため、抽出した「おはよう」に相当する崩れた異表記を人手でチェックした。チェックの基準としては、人が見て「おはよう」に関する文字列が含まれてい

*1 Twitter クライアントとは第三者が作成した Twitter 用のソフトウェアの事である

表 3: 使用したモデルとその結果

モデル	Precision	崩れた異表記の種類
ベースライン	1	51
長さ 1 の部分文字列	.731	142
長さ 2 の部分文字列	.953	61
長さ 3 の部分文字列	.405	25

表 4: 長さ 2 の部分文字列を用いた際に抽出した崩れた異表記の例

正しく抽出できた例
オハよう
おはほもーーーーーーーーーーーーーーーーーー
おはよーっ

誤って抽出した例
あああああああああああああああ
あああああああああああああああああああ
おおおおおおおおおおおおおおおおおおおお

るならば正解とした。具体的には

$$\text{Precision} = \frac{\text{正しく抽出した崩れた異表記の種類}}{\text{抽出した崩れた異表記の種類}}$$

である。

崩れた異表記を抽出した後に崩れルールを抽出するため、崩れルールが誤検出されないよう高い Precision を保ちつつ、より多くの種類の崩れた異表記を抽出することが重要である。

5. 考察

5.1 「おはよう」の崩れた異表記抽出に対する考察

表 3 に崩れた異表記の抽出結果を示す。表 3 では、ベースラインを用いて抽出した「おはよう」の崩れた異表記の種類は、実際より多い。なぜならば、ベースラインで抽出した文字列には、CRF の「おはよございます」の「ございます」部分に当たる「おはよご」が含まれていたが、「おはよう」に相当する文字列が含まれているため、正解にしているからである。一方、CRF においては「おはよございます」という入力に対して、「おはよ」部分しかされていないよう訓練されているため、ベースラインよりは抽出した「おはよう」の崩れた異表記の種類が少なくなっている。

長さ 1 の部分文字列のみを素性として使用した時、194 種類中 52 種類 (26.8%) を誤検出した。誤検出した種類の中に「お」もしくは半角の「オ」を 3 文字以上含む種類が 15 種類あり、「お」に対する重みが高い事が誤検出の原因であると伺える。しかし、カタカナを含む 9 種類の崩れた異表記や小書き文字を含む「ぉはょぅ」等の崩れた異表記を 30 種類抽出できた。その他、「ー」、「お」、「は」等が繰り返されていた異表記を抽出でき、より多くの種類の崩れた異表記を抽出できた。

表 4 に抽出した「おはよう」の崩れた異表記の例を示す。長さ 2 の部分文字列のみを素性として使用した時 64 種類中 3 種類 (4.7%) を誤検出した。しかし、長さ 1 の部分文字列を素性として使用した時と異なり、5 種類のカタカナを含む崩れた異表記と小書き文字を含む崩れた異表記は 5 種類抽出できた。その他、「ー」が繰り返されていた崩れた異表記は多数抽出できた。

なお、長さ 3 の部分文字列のみを素性として使用した時、42 種類中 17 種類 (40.5%) を誤検出した。誤検出した種類の内 9 種類は「あ」が 7 回以上繰り返した文字列を部分文字列として含んでいた。また、5 種類は 1 文字の記号であった。長さ 2 の部分文字列を素性として使用した時と比較すると、誤検出の原因は素性が疎になったことにより、抽出の精度が低くなったからだと考えられる。

表 5: 上位 5 崩れルールと各崩れルールの出現回数

抽出した崩れルール	出現回数
“う” → “ー”	361
“よう” → “”	285
“う” → “”	244
“” → “ー”	19
“” → “っ”	15

5.2 崩れルール抽出に対す考察

表 5 に長さ 2 の部分文字列を使用した結果に対して、抽出した崩れルールを示す。361 回出現した崩れルールである “う” → “ー”の内 339 回は「おはよー」という崩れた異表記を CRF が抽出したからである。なお、残りの 22 回は「おっはよー」等の「おはよー」と似た文字列より抽出された。“う”→ “ー” は JUMAN7.0 に組み込まれた崩れルールの 1 つ [笹野 12] であることから、崩れルールとして妥当だと判断できる。また “お” → “” という崩れルールが 10 回抽出できたが、その内 2 回は「オハヨ」という崩れた異表記を抽出したからであり、“お” → “” と “はよう” → “オハヨ” の 2 つの崩れルールが適用されたからである。直感的には “はよう” → “ハヨウ” という崩れルールが抽出されるべきであり、今後カタカナに変換する時のコストの調整等をして改善する。

6. 結論

本稿では「おはよう」の崩れた異表記と崩れルールを抽出した。実験の結果、素性に使用する部分文字列の長さにより、Precision と抽出できた崩れた異表記の種類の数との間に、トレードオフが存在することがわかった。崩れルールの抽出を視野に入れるなら、精度を保ちつつより多くの種類を抽出できる、長さ 2 の部分文字列を素性として扱った方が良い。ベースラインとして用いた、「おはよう」との編集距離 1 の部分文字列を取得するより、より多くの種類を抽出でき、かつ崩れルール抽出の際に誤検出が際立たない。

今後の課題としては 3 点挙げられる。まず「おはヨーグルト」等の音韻を意識している表現や口語的表現が多いため、音韻を素性として取り入れ、高い精度を保ちつつより多くの種類の崩れた異表記を抽出できるように改善する。また抽出した崩れルールを他の単語への適用し、他の単語の崩れた異表記を学習データなしに抽出する。それに加え、抽出した崩れルールと極性もしくは感情との対応付けをすることである。本稿で抽出した崩れルールである、“う”→ “ー”の内 339 回は、「おはよー」が部分文字列として含まれているツイートより、抽出されていた。その内約 66.7%に当たる 226 回は、正の極性を連想する顔文字と共起しており、“う”→ “ー”という崩れルールに正の極性を付けられる、と推測できる。

参考文献

[Brody 11] Brody, S. and Diakopoulos, N.: Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!! Using Word Lengthening to Detect Sentiment in Microblogs, in *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 562–570, Edinburgh, Scotland, UK. (2011), Association for Computational Linguistics

[Han 11] Han, B. and Baldwin, T.: Lexical Normalisation of Short Text Messages: Makn Sens a #twitter, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 368–378, Portland, Oregon, USA (2011), Association for Computational Linguistics

[Liu 11] Liu, F., Weng, F., Wang, B., and Liu, Y.: Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-categorization nor Supervision, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 71–76, Portland, Oregon, USA (2011), Association for Computational Linguistics

[Matsumoto 11] Matsumoto, K., Konishi, Y., Sayama, H., and Ren, F.: Analysis of Wakamono Kotoba Emotion Corpus and Its Application in Emotion Estimation, *International Journal of Advanced Intelligence*, Vol. 3, No. 1, pp. 1–24 (2011)

[Okazaki 07] Okazaki, N.: CRFsuite: a fast implementation of Conditional Random Fields (CRFs) (2007)

[笹野 12] 笹野 遼平, 鍛治 伸裕 : 不自然言語処理-枠に収まらない「リアルな」言語処理- : 2. 新しい語・崩れた表記の処理, 情報処理, Vol. 53, No. 3, pp. 211–216 (2012)

[水岡 11] 水岡 良彰, 鈴木 優 : マイクロブログを用いた感情表現収集, 第 10 回情報科学技術フォーラム (FIT2011) 講演論文集, pp. 291–294 (2011)

[前田 12] 前田 裕, 遠藤 勉, 嶋田 和孝 : 文体の違いを考慮したマイクロブログの極性判定, pp. 1110–1113 (2012)

[藤沼 12] 藤沼 祥成, 横野 光, Martínez-Gómez, P., 相澤 彰子 : Recognizing Variations of Japanese “Good Morning” Phrases in Twitter (言語理解とコミュニケーション・第 4 回集合知シンポジウム), 電子情報通信学会技術研究報告: 信学技報, Vol. 112, No. 367, pp. 35–40 (2012)