web上のテキストからの表記揺れ語獲得

斉藤 いつみ　貞光 九月　浅野 久子　松尾 義博
NTT メディアインテリジェンス研究所
{saito.itsumi, sadamitsu.kugatsu, asano.hisako,
matsuo.yoshihiro}@lab.ntt.co.jp

1　はじめに

ソーシャルメディアテキストは情報抽出や評判分析などの重要な情報源となっている．しかし，ソーシャルメディアテキストには，新語や，口語表現などの未知語が多く出現し，新聞等に比べ解析誤りが多く発生することが問題となっている．このような未知語問題についてはさまざまな未知語獲得や解析の研究が行われてきた．たとえば，単語の語幹と，後続するひらがな列の分布から語幹の区切りと品詞を推定する手法 [12] や，wikipedia などの既存のリソースを用いて辞書を構築する方法が知られている．これらの手法は，形態素辞書に存在しない単語の獲得に注力したものであり，形態素解析そのものの改善に効果を示すものである．一方最近では，口語調や小文字化，長音化，ひらがな化，カタカナ化など新聞等で用いられる標準的な表記から逸脱した崩れた表記揺れ語を正規化して解析するという研究が増加している [1, 2, 6]．これらの研究は，形態素解析精度の向上のみならず表記揺れのまとめあげなどの用途で効果を発揮するものである．特にソーシャルメディアテキストにおいては,「たっのしー」「きれぃ」など辞書語の崩れた表記が頻繁に利用されるため，表記揺れ語の頑健な解析や正規化の研究の重要度は増しており，日本語における表記正規化の研究も増加している [11, 9, 4, 8]

本研究では，このような崩れた語を含む表記揺れ語を web 上のテキストから自動的に獲得する手法を提案する．日本語において，これまでの崩れた表記を含む表記揺れ語の解析では，表記揺れのパタンについてはルールや学習データに基づいて生成するものがほとんどであった．これは，日本語が分かち書きされない言語であるため，分かち書きされていないテキスト中から未知の表記揺れ単語の区切りと対応する正規語を推定するということが困難な課題であったためと考えられる．英語では，非アノテーションデータから表記揺れ語と正規語の抽出を自動的に行う研究が存在する [2, 3] が，これらの手法は単語区切りが明示的であることを前提としており，直接日本語に適用することは難しい．

本論文では，web の短い文やフレーズに着目し，これらの単位ごとに類似度を計算して対応する正規の表記を推定することにより，特定のパタンに依存せずに web 上のテキストから表記揺れ語と正規語のペアを抽出する．類似度の指標としては音的な類似性と意味的な類似性を使用し，さらに類似度グラフを用いて周辺の複数の類似候補から正規語を推定することにより，高い精度で抽出を行うことができた．

本研究の貢献は大きく次の二点である．一点目は，web 上の短文・フレーズに着目し，形態素区切りが非明示的な対象についても精度よく表記揺れと対応する正規語のペアを抽出する方法を提案したこと，二点目はグラフ伝播法を用いることにより，抽出精度を大きく低下させずに獲得の再現率を上げることが可能であることを示したことである．本提案手法を用いて得られた候補を実際に形態素解析辞書に追加し，悪影響を抑えつつ精度を向上できることを確認した．

2　表記揺れ語の解析と本研究の対象

日本語における表記揺れ語解析の課題は，大きく 1) 表記揺れ辞書・パタンの獲得，2) 表記正規化と形態素解析の同時解析に分けることができる．1) では，カタカナ表記揺れや漢字略語など特定のパタンについては一定の成果が得られているものの，ソーシャルメディアテキスト上で多く現れるようなひらがな混じりの崩れ語を含む表記揺れ語の獲得はされていない．2) に関しては，表記揺れのパタンは既知（ルールや学習データから獲得）とし，形態素解析精度の向上を確認している．1) と 2) の課題は独立と考えることができ，さらなる精度向上のためには表記揺れパタン（辞書）の拡大が必要であることが言及されている [4, 8]．今回本論文が提案する手法は，既存システムが解析できない未知の表記揺れパタンを事前の知識に依存せずに獲得するものである．そのため，表記揺れ辞書の拡充などに有用であるとともに，既存の解析手法と組み合わせて利用することで，さらなる精度向上や時間経過による表記揺れ語の変化への対応が見込めるものである．

本研究が対象とする表記揺れ語は，縮約（行こう→行こ），長音化（たのしい→たのしー），小文字化（ありがとう→ぁりがとう），口語的音変化（すごい→すげえ），字種変化（バイト→ばいと）など発音の類似性が存在する語である．従来研究によって，ソーシャルメディアテキストの中で最も多い表記揺れ語が発音の類似性がある語であるとされており，日本語でも同様の傾向がみられるためである [4, 8]．また今回の手法では，事前に表記揺れのパタンを与えないため，従来研究が対象外としていた誤字（ありがとう→あらがとう）や略語（サムネイル→サムネ）なども，発音が近いものに関しては一部獲得することができる．

本研究で獲得する対象は，表記揺れ語と対応する正規語（辞書に存在する語）のペアである．辞書に存在する語を正規語として推定するため，獲得した表記揺れ語を形態素解析辞書として用いる場合には正規語の品詞をそのまま使用することができる．

3 提案手法

3.1 提案手法の全体像

提案手法では，まず 2 章で述べたように Twitter の短文･フレーズに着目し，粗く区切った分割テキストを生成する．その後分割済テキストの分割文字列をノードとみなし，各ノード間で類似度グラフを作成する．このグラフ上でラベル伝播法を用いて最も類似性の高いノードを推定することにより，表記揺れ語と対応する正規語のペアを抽出する．

3.2 表記揺れ語候補と分割テキストの生成

グラフ表現を用いて表記揺れ語を抽出するためには，まず表記揺れ語が含まれるような分割コーパスを作成する必要がある．しかし，今回獲得したい表記揺れ語は辞書に存在しないため，既存解析器でテキストを解析させると正しい区切りを抽出することができない．表記揺れ語候補を生成するため，本研究では Twitter の短文に着目した．Twitter では，文字数制限があることなどサービスの性質上短い文が多く,「おっはよーう！」や「みてみたぁーい」など単語単体や数単語からなるフレーズのみでつぶやかれるツイートも多く存在することが観測された．そのため，これら短い文（フレーズ）を単語区切り推定手がかりとして使用した．具体的には，Twitter の大規模テキストを指定した区切り文字（改行文字，句読点，記号，顔文字，スペース）で分割し，文字数が指定文字以下の文字列を表記揺れ語候補として抽出した．区切り文字を上記のよう設定することにより，最小単位か否かは不明確であるが少なくとも両端には形態素区切りが存在するような文字列を抽出することができる．今回は，10 文字以内の文字列を候補として抽出した．

次に，抽出した表記揺れ語を既存の辞書に追加し Mecab を用いてテキストを解析させた．この結果，もとの解析器では “おっ/は/よー/う/！” と解析されていた文が “おっはよーう” とまとまって解析されるなど，分割結果に正しい表記揺れ候補も多数含まれる分割テキストが生成された．ここで，抽出方法の特性から，単語のみならず「たのしかったぁぁ」「うっれしいよぉ」などの短いフレーズも一単語の候補として列挙される．本提案手法では，この時点では区切りを明示的に推定することをせず，短いフレーズもそのままノードとして扱うこととする．

3.3 類似度グラフを用いた正規語推定

3.3.1 類似度関数の定義

類似度グラフを生成するため，2 つのノード w_i， w_j 間の類似度を定義する必要がある．本提案手法では主要な指標として意味類似度と音類似度の 2 つを用いる．この 2 つを用いた理由は，音類似性が高くても意味が全く異なるものも存在するため，意味類似度を同時に用いることで高精度に同義ペアを抽出するためである．以下，2 つの指標の計算方法について詳細を示す．

意味類似度

ノード w_i と w_j 間の意味類似度については，word2vec を用いて計算を行った[1]．word2vec は，各ノードをベクトル空間に写像するもので，ノード間の文脈類似性を考慮することができる [7]． word2vec によって求めた各ノード間のベクトルのコサイン類似度を用いて，意味類似度 $semsim(w_i, w_j)$ を次のように定義する．

$$semsim(w_i, w_j) = cos(vec(w_i), vec(w_j)).$$

音類似度

音的類似度を計算するため，まずノードの表記を読み（アルファベット）に変換する．次に，音的な表記揺れにおいては母音の変化が特におこりやすいことから，母音の置換と削除を無視した編集距離である Modified Edit distance を計算する ($MED(w_i, w_j)$)．音類似度は，$MED(w_i, w_j)$ を用いて次のように定義した． $psim(w_i, w_j) = MED(w_i, w_j)/OC(w_i, w_j)$, ここで，$OC(w_i, w_j)$ は w_i と w_j の読み表記間の編集距離計算における全操作数を表し，単語（読み）長で正規化するための値である．

ノード間類似度

最後に，w_i と w_j 間の類似度を 2 つの指標を用いて次のように定義する．

$$W_{w_i w_j} = semsim(w_i, w_j) \cdot psim(w_i, w_j) \qquad (1)$$

ここで，自ノードとの類似度 $W_{w_i w_i}$ は 1 とした．

3.3.2 類似度グラフの構築

2.2 で作成した分割済コーパス中に出現するすべての分割文字列をノードとし，類似度グラフを構築する．類似度グラフの構築にあたり，ノード間類似度を用いて各ノードに対して近傍ノード（リンクを張るノード）集合 L の設定を行う．計算量の削減と情報伝播によるノイズの軽減のため，近傍リストには正規語候補として可能性が高い候補のみを設定する．今回は下記に示す条件をすべて満たすノードを，各ノードの近傍ノードリストとして設定した．

- 着目ノードの表層が正規ノードの条件を満たさない場合，または，意味類似度 $semsim(w_i, w_j)$ が閾値以上かつノード表層のテキスト中での出現頻度が着目ノードの出現頻度よりも大きい場合
- 着目ノードとのノード間類似度が閾値以上の場合
- 着目ノードとの読みの標準編集距離が閾値以内の場合

ここで，正規ノードとは次の 2 条件のいずれかを満たすノードとした．1) 表記が辞書に存在する，2) 既存解析器で解析させた結果が，形態素正解付きコーパスである BCCWJ に出現する単語の並びである．

2 ノード w_i， w_j 間の接続確率は，前項で定めた類似度指標 $W_{w_i w_j}$ を近傍リスト内のノードで正規化し

[1] http://code.google.com/p/word2vec/

た次の値を設定する．

$$p(w_j|w_i) = W_{w_i w_j}/\Sigma_{w_j} W_{w_i w_j} \quad (2)$$

上記の条件を満たす候補数があらかじめ定めた上限値に達するまで近傍リストに追加した．今回の実験では上限値を 5 とした．ここで，近傍リスト外のノードとの接続確率は 0 とする．そのため，類似度グラフはスパースなグラフとなる．

3.3.3 正規語推定

本項では，$P(v|w)$ の推定方法について記述する．ここで，w は各ノード（単語またはフレーズ）を表し，v は w の正規形を表す．推定方法は，Szummer ら [10] らによって提案されたラベル伝播法を参考に，ノード w_i を観測した際にその正規語が v である確率 $P(v|w_i)$ を近傍ノードとの接続確率を用いて次のように定義する．

$$P(v|w_i) = \Sigma_{w_j} P(v|w_j) p(w_j|w_i) \quad (3)$$

ここで，$p(w_j|w_i)$ は隣接ノードとの接続確率を表し，一定である．$P(v|w_j)$ は EM アルゴリズムを用いて求める．E-step では，$P(w_j|w_i, \tilde{v})$ を前ステップの $P(\tilde{v}|w_j)$ を用いて次のように計算する．

$$P(w_j|w_i, \tilde{v}) \propto P(\tilde{v}|w_j) p(w_j|w_i) \quad (4)$$

M-step では，$P(v|w_j)$ を次の式に基づいて更新する．

$$P(v|w_j) = \Sigma_{w_i:\tilde{v}=v} P(w_j|w_i, \tilde{v}) / \Sigma_{w_{i'}} P(w_j|w_{i'}, \tilde{v}) \quad (5)$$

$P(v|w_j)$ の初期値に関しては，近傍リストのうち正規ノードと自ノードのみを用いて接続確率を正規化した値を用いた．また，繰り返し計算の結果もっとも確率が高くなった候補を正規語候補とし，w と v のペアを抽出した上で，再度 2 単語間の音類似性，読み類似性，表層一致度を用いてフィルタリングを行った．

4 実験

提案手法の効果を確認するため，Twitter データを用いて実験を行う．評価は，実際に獲得できたペアの正解率や形態素解析に導入した際の形態素解析精度によって行う．

4.1 実験データ

表記揺れ語を抽出するため，表記揺れ語が多く存在していると思われ，かつ少ない単語数で構成される文が多い Twitter のデータを用いる．2013〜2014 年のクロールデータをランダムにサンプリングし約 7000 万ツイートを用いた．形態素解析器は Mecab[5] を使用し，ベースの形態素辞書は unidic 辞書を使用した．

4.2 結果

4.3 表記揺れ語の獲得結果

構築された類似度グラフのノード数は約 47 万ノードであり，これらについて正規語の確率を計算した．表 1 には，獲得された表記揺れ語の正解率と規模を示した．正解率は，獲得されたペアの中からランダムに

手法	正解率（50 サンプル）	獲得ペア数
ラベル伝播なし	0.98	13741
ラベル伝播あり	0.92	30483

表 1: 獲得された表記揺れ・正規語ペアの正解率・規模

表記揺れ語 w（正規語 v）	$p(v\|w)$	表記揺れ語 w（正規語 v）	$p(v\|w)$
うーれしい（嬉しい）	0.735	カコイイ（かっこいい）	0.667
うっれしー（嬉しい）	0.728	たでぃーま（ただいま）	1.00
うれしーっ （嬉しい）	0.517	ただーみゃ（ただいま）	0.987
嬉しひ （嬉しい）	0.678	きゃんわいい（かわいい）	0.649
うれすぃ （嬉しい）	0.542	くぁわいい（かわいい）	1.00

表 2: 提案手法によって推定された表記揺れ語例

50 ペアをサンプリングして人手で正しいペアか否かの判定を行った．ここで,「伝播なし」とは，前節で設定した隣接確率 $p(w_j|w_i)$ が最も大きい正規ノードを正規語として推定するものである．この場合，近傍リスト内に正規ノードが存在しない場合は正規語を推定することができない．表 1 の結果から，グラフ上の類似度グラフを用いた確率伝播の手法を用いることで，多少の精度低下はあるものの 2 倍以上獲得数を増やすことができていることがわかる．また伝播なしでは，獲得精度は高く見えるものの，実際に獲得された候補が長音の挿入や促音の挿入など比較的変化の少ない表記揺れ語が多いのに対し，伝播ありではより多様な候補が獲得できている．

表 2 には提案手法（伝播あり）によって獲得した表記揺れ語の例を記載した．括弧内に示しているのが，推定された正規語である．表 2 の結果からもわかるように，われわれの手法はよく知られているパタン（長音化，小文字化等）のほかにも，これまでの研究においてルールとして報告されていないような多様な表記揺れパタンも自動的に獲得できていることがわかる．獲得したバリエーションから，単語に依存せずにおこる表記揺れパタンに関しては他の同一品詞の単語等にも表記バリエーションのパタンを展開することによりさらにカバー率が向上すると期待できる．一方，単語ごとに表記揺れパタンの獲得数には差があり，例えば感動詞「おはよう」は表記揺れ語の獲得数が約 400 と非常に多く，単語特有のバリエーションも多かった（おはろー，おぱよん，おはー，おはやっぷー，おはもー，おはぬー等）．このような単語特有のバリエーションは単語ごとに分布が異なるため，自動的に獲得できることのメリットが特に大きいと考えられる．さらに,「おめでとう→おでめとう」のような誤字や「アンソロジー→あんそろ」のような略語など，従来研究では扱っていなかったような表記バリエーションも獲得することができた．

表 3 にはフレーズレベルの崩れ語獲得例を示した．正規形には，形態素境界を “/” で表している．今回の手法は，明示的に単語区切りを正しく求めてから正規形を推定するのではなく，初期分割候補としては粗い区切りを生成し，その各分割文字列に対して正規形を求める手法となっているため，このような多対多の対

表記揺れ語 w	正規形 v	$p(v\|w)$
おめっとう	お/めでとう	0.959
おねげーします	お/ねがい/します/	1.0
ちょーかわええ	超/かわいい /	0.651
つかりたーー	疲れ/た	1.0

表 3: 提案手法によって獲得されたフレーズレベルの表記揺れ語獲得例

応関係も獲得することができる．ソーシャルメディア上の表記揺れ語には，形態素レベルで一対一の対応関係がつくものばかりではないが，このような多対多の対応関係については，既存研究でも扱われていなかった．日本語形態素解析においては，短いひらがな等をむやみに辞書に入れると解析結果が悪化することが知られており，表記正規化においてもあるフレーズでまとまって起きるような表記揺れに関しては，複数形態素をまとめて変換するなどの方法も有効であると考えれる．

獲得エラーの例としては,「おかえり→おかあり」や「コスプレ→コスプリ」があった．このような例は，意味的にも音的にも近い語であるため，より精緻な識別を行わなければ識別が難しいものと思われる．また，「良いね→良いのう」など，フレーズレベルで近い意味・音のものが紐づけられてしまったものも散見された．これらは，文法的な情報も使うと識別が可能になると考えられる．

4.4 形態素解析への導入実験結果

得られた表記揺れ語を形態素解析辞書に入れ，形態素解析に与える影響を評価した．このような表記正規化の研究においては，表記正規化を行うことによって崩れたテキストの解析結果がよくなるかだけではなく，整ったテキストの解析結果が悪化しないか，という観点も重要となる．そこで本研究では，比較的整ったテキストである BCCWJ による評価と，崩れた表記が含まれる Twitter テキストの両方で評価を行った．今回は簡単のため，一形態素のみからなる獲得ペアについて，辞書化を行った．辞書化にあたっては，2 文字以下のひらがな単語は解析悪化につながるため，3 文字以上の単語に限定し，辞書に存在する表記の単語は削除するなどの処理を行い，24955 個のエントリを生成した．この辞書をベース辞書に追加し，まず BCCWJ のランダムサンプリングデータで単語区切り＋品詞の F 値を計測したところ，いずれも 0.989 となり精度はほぼ同等となった．

次に，Twitter テキストを表記揺れ辞書あり，なしでそれぞれ解析し，ランダムに選んだ 100 箇所について解析差分の人手チェックを行った．評価は，A：改善，B：不変，C：悪化の 3 種類に分類した．その結果それぞれの観測回数は，A：65，B：18，C：17 となり，改善がその他を大きく上回る結果となった．改善例としては，狙い通り口語調などの表記揺れを吸収できたものがあり，例えば「きっ/たく/ー」→「きったくー（帰宅）」や,「ネム/ゥ」→「ネムゥ（眠い）」などがあった．改悪例としては,「そろそろ/お/願い」→「そろそろお/願い/」などのひらがな機能語が結合してしまうものや,「ウーーーーン」→「ウーーーー/ン」など，表記揺れ辞書の追加によるカタカナの過分割があった．このような例に関しては，表記揺れ語の獲得手法をより洗練させるとともに解析手法側での工夫も検討していきたい．

5 おわりに

提案手法は，少ない事前知識で表記揺れ語を高精度に獲得することができる．また，獲得した候補を既存手法に組み込むことも可能である．今回は獲得できなかった音類似性の低い略語の獲得等も，類似度関数の改良により可能と考えられるため，今後の課題としたい．

参考文献

[1] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pp. 368–378, 2011.

[2] Bo Han, Paul Cook, and Timothy Baldwin. Automatically constructing a normalisation dictionary for microblogs. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 421–432, 2012.

[3] Hany Hassan and Arul Menezes. Social text normalization using contextual graph random walks. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1577–1586, August 2013.

[4] Nobuhiro Kaji and Masaru Kitsuregawa. Accurate word segmentation and pos tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proceedings of the 2014 Conference on EMNLP*, pp. 99–109, Doha, Qatar, October 2014. Association for Computational Linguistics.

[5] T. KUDO. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.net/*, 2005.

[6] Chen Li and Yang Liu. Improving text normalization using character-blocks based models and system combination. *Proceedings of COLING 2012*, pp. 1587–1602, 2012.

[7] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[8] Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. Morphological analysis for japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1773–1782, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[9] Ryohei Sasano, Sadao Kurohashi, and Manabu Okumura. A simple approach to unknown word processing in japanese morphological analysis. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 162–170, 2013.

[10] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, pp. 945–952. MIT Press, 2002.

[11] 工藤拓, 市川宙, David Talbot, 賀沢秀人. web 上のひらがな交じり文に頑健な形態素解析. 自然言語処理学会年次大会講演集, 2012.

[12] 村脇有吾, 黒橋禎夫. 形態論的制約を用いたオンライン未知語獲得. 自然言語処理, Vol. 17, No. 1, pp. 55–75, 2010.