# 語彙正規化との同時処理による高精度な形態素解析

# Accurate Morphological Analysis by Jointly Performing Lexical Normalization

鍛治　伸裕[1*]　喜連川　優[1,2]
Nobuhiro Kaji[1]　Masaru Kitsuregawa[1,2]

[1] 東京大学 生産技術研究所
[1] Institute of Industrial Science, The University of Tokyo
[2] 国立情報学研究所
[2] National Institute of Informatics

**Abstract:** Microblogs have recently become an important target of natural language processing. Since microblogs offer an instant way of posting textual messages, there are emerging possibilities that social behaviors can be monitored in real-time by analyzing text data on microblogs. However, the abundant use of ill-formed words on microblogs makes accurate text analysis, especially morphological analysis in Japanese case, difficult. We address this problem by proposing a joint model of lexical normalization and morphological analysis.

## 1　はじめに

近年，Twitter[1]に代表されるマイクロブログは，自然言語処理技術の適用対象として急速に重要性を増しつつある．即時的な情報発信を特徴とするマイクロブログには，様々な情報がリアルタイムで書き込まれる．そのため，自然言語処理技術を活用してマイクロブログの内容を分析することにより，実世界で「今」発生しているイベントを検知することや，ユーザ経験をリアルタイムで抽出し共有することなどが，実現可能になりつつある．

マイクロブログ上では「すんげぇ〜」「悪ぃ」「らいぶ」など，崩れた表記で綴られた単語 (**崩れ表記語**と呼ぶ) が頻繁に使われる．そのため，整った表記の単語 (**整表記語**と呼ぶ) が多数を占める新聞コーパスなどから学習される従来の解析器は，マイクロブログを頑健に解析することが困難となっている．日本語のように単語を分かち書きしない言語においては，形態素解析の精度が大きく低下してしまうことがとりわけ大きな問題となる．

本論文では，マイクロブログに対する高精度な形態素解析処理を実現するため，語彙正規化と形態素解析を統合した解析モデルを提案する．崩れ表記語を正規化して得られる整表記語の情報を，形態素解析モデルが利用可能とすることによって，マイクロブログの形態素解析精度を向上させることが我々の狙いである．提案モデルの特徴は，大規模な言語モデルを使って，正規化後のテキストの自然さを素性として利用できる点である．実験では，このような従来の形態素解析では考えることができない素性を用いることによって，解析精度が大きく向上することを示す．

*連絡先： 東京大学 生産技術研究所
東京都目黒区駒場 4-6-1
E-mail: kaji@tkl.iis.u-tokyo.ac.jp

[1]https://twitter.com

## 2　語彙の正規化

語彙の正規化という研究は歴史が浅い分野である．そのため，現在のところ，そのタスク設定に関して，研究者の間に明確な合意形成が行われているとは言い難い．そこで，本節では，本研究が取り扱う正規化タスクの設定を明確化するための議論を行う．

### 2.1　本研究が扱う崩れ表記語

まずはじめに，本研究において取り扱う崩れ表記語について議論を行う．

整表記語から崩れ表記語が生成される過程には，音韻的な要素が大きく関わっていることが，過去の語彙正規化に関する研究において指摘されている．例えばXia ら [15] は，中国語のチャットテキストで使われる崩れ表記を調査した結果，99%以上の崩れ表記に対して，整表記との間に音韻的な対応関係が存在していたことを報告している．また，文献 [14] では，発音の類似性

表 1: 本研究が扱う崩れ表記語の例．それぞれの崩れ表記語を整表記語に正規化したものを括弧内に併記している．

| |
|---|
| 英語 分かんねぇ(分からない) |
| 取ら れとる(れている) |
| 逮捕者 きたーー(きた) |
| かわいいい(かわいい) |
| 周囲の目が キニナリマス(気になります) |
| 悪ぃ(悪い) かよ |

に基づいて生成された崩れ表記語が，英語ツイートにおいて最も数が多かったことが報告されている．

こうした従来研究における調査結果を踏まえ，本研究でも，音韻的な要因によって生成された崩れ表記語を正規化の対象とする (表 1)．具体的には，縮約，長音化 [2]，方言，字種変化 [17] などを取り扱う．これらは，他言語における語彙正規化の研究が扱っているものとほぼ同様であるが，日本語の場合，字種変化によって生成される崩れ表記語が多いことが特徴となる．英語における字種変化は，アルファベットとアラビア数字や記号の間の転写に限定されるため (例: *for* と 4 など)，その種類数は少ない．これに対して日本語では，アルファベット以外にも 3 種類の文字 (平仮名，片仮名，漢字) が同時に使われるため，字種間の転写は頻繁に行われる [4]．

### 2.2 形態素解析高度化のための語彙正規形

語彙の正規化とは，単純には，入力テキスト中の崩れ表記語を整表記語に変換する処理であると考えることができる．以下では，テキストで実際に観測される語形のことを**表出形**，それを正規化したものを**正規形**と呼ぶ．

しかし，上記のような単純な定義では，Eisenstein[3] も指摘しているように，崩れ表記語に対応する整表記語が複数存在する場合，正規形が一意に定まらないことが問題となる．例えば，崩れ表記語「クダケタ」の整表記語には「くだけた」と「砕けた」の 2 つが存在するが，そのどちらが正規形として相応しいのかを決めることは難しい．同様の問題は，*flvr* の正規形を *flavor* とするのか，それとも *flavour* とするのかなど，英語においても発生する．

この問題に対する我々の考え方は以下のようなものである．まず，我々が語彙の正規化を行う目的は，形態素解析の精度を向上させることである．そのため，語彙正規化が果たすべき最も重要な役割は，整表記語だけを使って記述された，形態素解析が容易なテキストを生成することであると考える．一方，語彙の正規化には単語表記を統一して疎データ問題を緩和するいう側面もあるが，本研究では，それはあくまでも副産物であるという立場を取り，正規化の主たる目的とは考えない．

こうした考えに基づき，ある崩れ表記語に対する整表記が複数存在する場合には，その全てが適格な正規形であるとし，崩れ表記語をそのいずれかに変換する処理のことを語彙正規化と呼ぶ．我々のタスク設定においては，ある文脈では「クダケタ」が「くだけた」に変換され，別の文脈では「砕けた」に変換されるということもありうる．

また，単語表記を統一することは必ずしも重要ではないという考えから，整表記を別の整表記に変換するような処理 (例:「くだけた」から「砕けた」への変換) は，語彙正規化の対象外とする．整表記語については，表出形と正規形は必ず同一であるとする．

## 3 提案手法の概要

本節では，本研究が提案する語彙正規化と形態素解析の同時解析手法の概要を説明する．

従来，形態素解析処理を行う方法としては，単語ラティスの経路探索に基づく手法が広く用いられている [7, 8, 9]．これは，入力文に対する形態素解析結果の候補集合を，エッジに品詞タグ[2]がラベル付けされた単語ラティスを用いて表現し，識別モデルによって最適な経路 (=解析結果) を選択するという方法である．このアプローチの利点は，膨大な数の解候補を単語ラティスという圧縮構造を使ってコンパクトに表現することによって，豊富な素性を用いた識別モデルを効率的に学習できることである [7, 8]．これは forest reranking[6] と同じ考え方である．

本研究では，語彙正規化と形態素解析の同時処理においても，同様に単語ラティスの経路探索に基づくアプローチをとる．すなわち，エッジに対して，品詞タグ，単語の正規形，正規形に対する品詞タグの 3 つ組がラベル付けされた単語ラティスを構築し，識別モデルを用いて最適経路を選択することによって，語彙正規化と形態素解析を同時に行う．

ここで，単語表出形に対する品詞タグと，正規形に対する品詞タグの 2 種類がエッジに付与されることに注意されたい．以下では，これらを区別するため，表出形と正規形に対する品詞のことを，それぞれ**表出品詞**，**正規品詞**と呼ぶ．多くの場合，表出品詞と正規品詞は同一となる．しかし，日本語のように活用が豊富な言語では，崩れ表記語に個別の活用形が定義されていることがあるため，一般的には表出品詞と正規品詞は異なる．例えば，JUMAN 品詞体系では，推量形が

[2]本稿では品詞，活用型，活用形をまとめたものを便宜上，品詞と呼ぶことにする．

表 2: 正規化辞書の例．品詞活用体系は JUMAN 辞書に基づいているが，スペースの都合上，品詞情報は品詞大分類と活用形のみを記載している．

| 表出形 | 表出品詞 | 正規形 | 正規品詞 |
|---|---|---|---|
| すげえ | 形容詞 (基本形) | すごい | 形容詞 (基本形) |
| 戻ろ | 動詞 (省略推量形) | 戻ろう | 動詞 (推量形) |
| らいぶ | 名詞 | ライブ | 名詞 |
| キタナイ | 形容詞 (基本形) | きたない | 形容詞 (基本形) |

縮約された崩れ表記 (例: 戻ろ) の活用形は省略推量形となるが，その正規形 (例: 戻ろう) の活用形は推量形となる．品詞体系によっては正規品詞を導入する意味がないこともありうるが，そのような場合は，本論文で議論する場合の特殊形として扱うことができる．

単語ラティスの構築には，文字単位の統計モデル [8] と辞書の 2 つを利用したハイブリッドな方法を提案する．ここで言う辞書とは，単語の表出形，表出品詞，正規形，正規品詞の 4 つ組を列挙したリストのことである．このような辞書データのことを**正規化辞書**と呼ぶ (表 2)．

単語ラティスの最適経路を選ぶための識別モデルは，従来の形態素解析に使われていたものと類似しているが，以下の 2 点が異なる．まず，入力を正規化した結果が自然な日本語となるように，大規模な言語モデルに基づく素性として用いる．また，識別モデルの学習には，マイクロブログなどの崩れ表記の多いドメインのテキストに対して，形態素情報と正規化情報をアノテーションした大規模な学習コーパスが必要となる．しかし，そのような言語資源は入手することが難しいため，新聞記事に形態素情報のみがアノテーションされた既存コーパスから学習することを提案する．

以下，4 節と 5 節では，提案手法の詳細を述べる．まず，4 節では，正規化辞書を構築する方法を説明する．5 節では，正規化辞書を用いて単語ラティスを構築し，その最適経路を探索する方法について述べる．

# 4 正規化辞書の構築

本節では正規化辞書の構築方法を説明する．

これまで，正規化辞書として使えるような言語資源は十分に整備が行われていない．しかし，一方，単語分割や品詞タグ付与の研究においては，単語とそれが取りうる品詞の組を登録した辞書 (**品詞辞書**と呼ぶ) が広く用いられており，品詞辞書であれば大規模なものが容易に入手可能である．そこで，既存の品詞辞書を有効活用するということを考える．

既存の品詞辞書のエントリは概ね整表記語であり，崩れ表記語の数は少ない．そこで，まず既存の品詞辞書から人手で崩れ表記エントリを同定し，それらに適切な正規形と正規品詞を与えることによって，品詞辞書から正規化辞書を作成する．しかし，このようにして作成された正規化辞書には，十分な数の崩れ表記語は登録されていない．そこで，正規化辞書エントリ中の表出形 (= 整表記) を崩れ表記に自動変換することによって，新しいエントリを自動的に生成する．以下では，その変換方法について詳しく述べる．

## 4.1 整表記から崩れ表記への変換

整表記から崩れ表記への変換に関しては，これまでにもいくつかの方法が提案されている．それらは，人手で変換規則を作成する方法 [2, 12] と，ラベル付きコーパスから雑音のある通信路モデルなどの統計モデルを学習する方法 [14, 15] に大別することができる．しかし，後者のアプローチは大量のラベル付きコーパスが必要になることから，どのような言語でも簡単に適用できるわけではない [16]．英語については，ラベル無しコーパスから学習するという方法も盛んに研究されているが [5, 16]，それらは単語が空白で区切られていることを前提としているため，日本語に直接適用することは難しい．

そこで我々は，整表記から崩れ表記への変換規則を人手で記述する．紙面の都合上，実際に作成した変換規則を全て詳細に説明することは難しく，また，ここで用いる変換規則自体は本研究における主要な提案ではないため，ここでは規則の概要を説明するに留める．

変換規則は大きく 2 種類のものを作成する．まず 1 つ目は，特定の文字列を，発音が類似した別の文字列に変換するというものである．片仮名から平仮名への変換 (例: ツイッター → ついったー)，長母音から長音記号への変換 (例: そうじ → そーじ)，頻出する語尾変化 (例: すごい → すげえ) などを扱っている．もう 1 つの規則は長音化 (例: おめでとう → おめでとううう) を扱うためのものであり，文献 [12] と同様の規則に基づいて，任意個の母音または長音記号を挿入する．後者の変換規則は，正規化辞書のエントリを無限に増やしてしまうが，エントリ集合をオートマトンで表現することによって対応する．

# 5 語彙正規化と形態素解析の同時処理モデル

前節で説明した正規化辞書を用いて，語彙正規化と形態素解析を同時に行うモデルを構築する．具体的には，入力文 $x$ が与えられたときに，単語表出形列 $\boldsymbol{w} = (w_1, w_2 \dots w_n)$ と，それに対応する表出品詞列

$\boldsymbol{t} = (t_1, t_2 \dots t_n)$，正規形列 $\boldsymbol{v} = (v_1, v_2 \dots v_n)$，正規品詞列 $\boldsymbol{s} = (s_1, s_2 \dots s_n)$ を求める，以下のようなモデルを構築する．

$$(\boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s}) = \underset{(\boldsymbol{w},\boldsymbol{t},\boldsymbol{v},\boldsymbol{s}) \in \mathcal{G}(x)}{\arg\max} \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s}) \cdot \boldsymbol{\theta}$$

ここで $\mathcal{G}(x)$ は入力文 $x$ に対する解候補の集合であり，単語ラティスの形で与えられる．$\boldsymbol{f}(\boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s})$ は素性関数で，$\boldsymbol{\theta}$ は重みベクトルである．

## 5.1 単語ラティスの構築

単語ラティスの構築には，文字単位の統計モデル [8] と正規化辞書を利用したハイブリッドな方法を提案する．

まずはじめに，Kaji ら [8] と同様に，文字単位の単語分割モデルを構築し，その上位 $k$ 件[3]の出力を併合することによって単語ラティスを構築する．ただし，この段階ではエッジにラベルは付与されていないことに注意されたい．

次に，各エッジに対して，ラベル (= 表出品詞，正規形，正規品詞の 3 つ組) の集合を割り当てる．ラベル集合の割り当てには，正規化辞書に基づく方法とヒューリスティクスに基づく方法の 2 種類を用いる．正規化辞書だけでなくヒューリスティクスも用いるのは，正規化辞書に現れない未知語に対応するためである．

ラベル集合割り当ての具体的な手続きは以下の通りである．まず，エッジと同じ表出形を持つエントリを正規化辞書から検索し，そのエントリに含まれる表出品詞，正規形，正規品詞の 3 つ組を割り当てる．次に，オープンクラスの品詞集合をあらかじめ定義しておき，その中から，活用語尾が一致するなどの条件を満たすものを表出品詞とする．このとき，正規形と正規品詞は不明であるため，表出形および表出品詞と同じとする．

## 5.2 素性

我々の用いる素性は，文字素性，単語素性，単語 2-gram 素性，言語モデル素性の 4 種類に分類することができる．既存の形態素モデルとの大きな違いは最後の言語モデル素性である．正規形化されたテキストの自然さを，大規模な言語モデルを使って数値化し，それを素性とすることによって精度向上を実現する．以下ではそれぞれの素性について詳しく説明を行う．

文字素性のテンプレートを表 3 に示す．これは，文字単位の単語分割モデル [11] において用いられるものとほぼ同様である．各文字 $c$ に対して，その周辺に出現する文字 $n$-gram と，単語内における文字 $c$ の位置情報 (単語の先頭とそれ以外の 2 値) を組み合わせたものである．文字種 $n$-gram に対しても同様の素性を用いる．表 3 の最後の行は，Neubig らが提案した辞書素性である [11]．この素性は，入力文に含まれる文字列が辞書に単語として登録されている場合に発火し，その文字列と注目している文字の相対的な位置関係，およびその文字列の長さを符号化している．

単語素性のテンプレートを表 4 に示す．単語の正規形，正規形の単語長，正規形の接辞文字列，単語が正規化辞書に登録されているかどうかという情報を用いる．

単語 2-gram 素性のテンプレートを表 5 に示す．正規形 2-gram 素性[4]は，計算量の問題から学習コーパスに頻出する上位 1000 単語のものに限定する [11]．

言語モデル素性は，既存の形態素解析器で解析したラベル無しテキストを用いて，正規形 $n$-gram の自然さを数値化する．解候補に含まれる正規形 $n$-gram($n$=2, 3) の頻度に 1 を足したものの対数を素性値を用いる．ここで頻度に 1 を足しているのは，頻度が 0 となる $n$-gram に対応するためである．言語モデル素性を効率的に計算するため実装には Aho-Corasick 法 [1] を用いる．

表 4: 単語素性テンプレート．$v$ と $s$ は単語の正規形と正規品詞を表す．$\textsc{length}(v)$ は正規形 $v$ の文字数 (1, 2, 3, 4, 5≤) を返す関数である．$\textsc{prfix}(n, v)$ と $\textsc{suffix}(n, v)$ は，正規形 $v$ の長さ $n$ の接頭文字列と接尾文字列を返す関数である．$\textsc{dict}(w, t, v, s)$ は，表出形 $w$，表出品詞 $t$，正規形 $v$，正規品詞 $s$ の 4 つ組が正規化辞書に登録されているかどうかを表す関数である．

| 名称 | 素性テンプレート |
|---|---|
| 正規形 | $\langle v, s \rangle$ |
| 正規形長 | $\langle \textsc{length}(v), s \rangle$ |
| 接頭文字列 | $\langle \textsc{prefix}(1, v), s \rangle$, $\langle \textsc{prefix}(2, v), s \rangle$ |
| 接尾文字列 | $\langle \textsc{suffix}(1, v), s \rangle$, $\langle \textsc{suffix}(2, v), s \rangle$ |
| 辞書情報 | $\langle \textsc{dict}(w, t, v, s) \rangle$ |

表 5: 単語 2-gram 素性のテンプレート．ただし正規形 2-gram は，学習コーパスに頻出する上位 1000 単語の正規形に限定する．

| 名称 | 素性テンプレート |
|---|---|
| 正規品詞 2-gram | $\langle s_{i-1}, s_i \rangle$ |
| 正規形 2-gram | $\langle v_{i-1}, s_{i-1}, v_i, s_i \rangle$ |

## 5.3 潜在パーセプトロンによる学習

提案モデルのパラメータは，入力文 $x$ に形態素情報 $(\boldsymbol{w}, \boldsymbol{t})$ と正規化情報 $(\boldsymbol{v}, \boldsymbol{s})$ がアノテーションされたコーパス $\mathcal{C} = \{(x, \boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s})\}$ があれば容易に学習可能であ

[3]実験では $k$=16 とした．

[4]正確には「正規品詞と正規形の組」の 2-gram であるが，簡単のため正規形 2-gram と呼ぶ．これ以降，正規形 $n$-gram と言った場合も同様である．

表 3: 文字素性のテンプレート．$c_i$ と $c'_i$ は現在注目している文字およびその文字種を表す．文字種には (1) アルファベット，(2) 漢字，(3) 平仮名，(4) 片仮名，(5) 数字，(6) その他，の 6 種類を用いる．$b$ は単語内における $c_i$ の位置 (先頭とそれ以外の 2 値) を表す．$c_{i-1}$ や $c'_{i+1}$ は対象文字の周辺に出現する文字およびその文字種を表す．BEGIN(END) は，$c_i$ から始まる (の直前で終わる) 文字列が辞書に登録されていることを示す．INSIDE は，$c_i$ を内部に含む文字列が辞書に登録されていることを示す．$s$ は，そのときに辞書に登録されてる文字列の文字数 (1，2，3，4 または 5≤) である．

| 名称 | 素性テンプレート |
|---|---|
| 文字 $n$-gram | $\langle c_{i-1}, b\rangle$, $\langle c_i, b\rangle$, $\langle c_{i+1}, b\rangle$, $\langle c_{i-2}, c_{i-1}, b\rangle$, $\langle c_{i-1}, c_i, b\rangle$, $\langle c_i, c_{i+1}, b\rangle$, $\langle c_{i+1}, c_{i+2}, b\rangle$, $\langle c_{i-3}, c_{i-2}, c_{i-1}, b\rangle$, $\langle c_{i-2}, c_{i-1}, c_i, b\rangle$, $\langle c_{i-1}, c_i, c_{i+1}, b\rangle$ $\langle c_i, c_{i+1}, c_{i+2}, b\rangle$, $\langle c_{i+1}, c_{i+2}, c_{i+3}, b\rangle$ |
| 文字種 $n$-gram | $\langle c'_{i-1}, b\rangle$, $\langle c'_i, b\rangle$, $\langle c'_{i+1}, b\rangle$, $\langle c'_{i-2}, c'_{i-1}, b\rangle$, $\langle c'_{i-1}, c'_i, b\rangle$, $\langle c'_i, c'_{i+1}, b\rangle$, $\langle c'_{i+1}, c'_{i+2}, b\rangle$, $\langle c'_{i-3}, c'_{i-2}, c'_{i-1}, b\rangle$, $\langle c'_{i-2}, c'_{i-1}, c'_i, b\rangle$, $\langle c'_{i-1}, c'_i, c'_{i+1}, b\rangle$, $\langle c'_i, c'_{i+1}, c'_{i+2}, b\rangle$, $\langle c'_{i+1}, c'_{i+2}, c'_{i+3}, b\rangle$ |
| 辞書情報 | $\langle$BEGIN, $b\rangle$, $\langle$END, $b\rangle$, $\langle$INSIDE, $b\rangle$, $\langle$BEGIN, $s, b\rangle$, $\langle$END, $s, b\rangle$, $\langle$INSIDE, $s, b\rangle$ |

**Algorithm 1** 潜在パーセプトロンによる学習

1: $\boldsymbol{\theta} \leftarrow \mathbf{0}$
2: **for** $i = 1 \ldots m$ **do**
3:  **for** $(x, \boldsymbol{w}, \boldsymbol{t}) \in \mathcal{C}'$ **do**
4:  $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \leftarrow$ DECODING$(x, \boldsymbol{\theta})$
5:  $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{s}}) \leftarrow$ CONSTRAINEDDECODING$(x, \boldsymbol{w}, \boldsymbol{t}, \boldsymbol{\theta})$
6:  **if** $\boldsymbol{w} \neq \hat{\boldsymbol{w}}$ or $\boldsymbol{t} \neq \hat{\boldsymbol{t}}$ **then**
7:  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{t}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{s}}) - \boldsymbol{f}(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$
8:  **end if**
9:  **end for**
10: **end for**
11: **return** AVERAGE$(\boldsymbol{\theta})$

る．しかし，そのような学習コーパスは構築に要する作業コストが大きいことから，本研究では，形態素情報のみがアノテーションされた既存の新聞記事コーパス $\mathcal{C}' = \{(x, \boldsymbol{w}, \boldsymbol{t})\}$ をパラメータ学習に再利用することを提案する．

本論文では，正規形 $\boldsymbol{v}$ と正規品詞 $\boldsymbol{s}$ を潜在変数とみなし，潜在パーセプトロン [13] を使ってパラメータの学習を行うことを提案する．表出形 $\boldsymbol{w}$ と表出品詞 $\boldsymbol{t}$ は教師データとして与えられているため，正規化辞書を使うことによって，正規形 $\boldsymbol{v}$ と正規品詞 $\boldsymbol{s}$ の候補を効果的に絞り込むことができる．そのため，新聞記事コーパスに正規化情報をあらためてアノテーションしなくても，潜在変数モデルを使うことによって，良いモデルが学習されることが期待できる．

潜在パーセプトロンによる学習手順をアルゴリズム 1 に示す．まず，通常のパーセプトロンと同じく，パラメータ $\boldsymbol{\theta}$ をゼロベクトルに初期化する (1 行目)．そして，各学習事例 $(x, \boldsymbol{w}, \boldsymbol{t}) \in \mathcal{C}$ に対して，現在のパラメータの元でスコアが最大となる解 $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$ を求める (4 行目)．さらに，表出形と表出品詞が正解と等しくなるという制約のもとで，スコアが最大となるような正規形と正規品詞 $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{s}})$ を求める (5 行目)．そして，$\hat{\boldsymbol{w}}$ または $\hat{\boldsymbol{t}}$ のいづれかが正解と異なる場合には，$(\boldsymbol{w}, \boldsymbol{t}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{s}})$ を正解事例とみなして，パーセプトロンの更新式に従ってパラメータの値を更新する (6-8 行目)．そして，最後にパラメータの平均化を行う (11 行目)．

表 6: マイクロブログコーパスでの形態素解析精度 (単語単位の適合率，再現率，F 値) の比較．

| | 適合率 | 再現率 | F 値 |
|---|---|---|---|
| MeCab | 67.5 | 77.9 | 72.5 |
| 提案法 (言語モデル素性無し) | 70.9 | 76.5 | 73.6 |
| 提案法 (言語モデル素性有り) | **71.2** | **77.0** | **74.0** |

# 6 実験

提案モデルの学習に用いたデータは次の通りである．正規化辞書のもととなる品詞辞書は JUMAN 辞書 Version 7.0[5]を用いた．潜在パーセプトロンの学習には京都大学コーパス Version 4.0[10] を用いた．言語モデル素性の計算には，ウェブから収集した 20 億文のブログテキストを MeCab[6]で解析した結果を用いた．

手法の評価には，筆者が構築したマイクロブログコーパスを用いた [18]．このコーパスは，Twitter への投稿から無作為に抽出した 1831 文に対して形態素情報が付与されている．このコーパスを用いて形態素解析処理の F 値 [9] を求めることによって手法の評価を行う．比較のため，代表的な統計的形態素解析システムである MeCab と，言語モデル素性を用いなかった場合の提案手法の結果を併せて報告する．

表 6 に結果を示す．提案手法は MeCab の精度を大きく上回っており，提案手法によって，崩れ表記語の形態素解析を頑健に行うことが可能になっていることが確認できる．また，言語モデル素性が精度の向上に貢献していることも確認できる．

# 7 おわりに

本論文では，マイクロブログに対する高精度な形態素解析処理を実現するため，語彙正規化と形態素解析

[5]http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN
[6]http://code.google.com/p/mecab

を統合した解析モデルを提案した．そして，実際にマイクロブログにアノテーションした評価データを用いて，手法の有効性を確認した．

今後の課題としては，略語や誤字脱字など，現在扱っていない崩れ表記語の正規化があげられる．略語の場合は，単語から句への変換が必要になる場合もあるが，正規化辞書さえ作ることができれば，本論文で提案した方法の単純な拡張で扱える．そのため，正規化辞書の構築が技術的な課題になると考えられる．

## 謝辞

本研究は，最先端研究開発支援プログラム「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価」の支援を受けている．

## 参考文献

[1] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*, Vol. 18, No. 6, pp. 330–340, 1975.

[2] Samuel Brody and Nicholas Diakopoulos. Cooooooooooooooollllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs. In *Proceedings of EMNLP*, pp. 562–570, 2011.

[3] Jacob Eisenstein. What to do about bad language on the internet. In *Proceedings of NAACL*, pp. 359–369, 2013.

[4] Masato Hagiwara and Hisami Suzuki. Japanese query alteration based on lexical semantic similarity. In *Proceedings of NAACL*, pp. 191–199, 2009.

[5] Bo Han and Timothy Baldwin. Lexical normalization of short text messages: Makin sens a #twitter. In *Proceedings of ACL*, pp. 368–378, 2011.

[6] Liang Huang. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL*, pp. 586–594, 2008.

[7] Wenbin Jiang, Haitao Mi, and Qun Liu. Word lattice reranking for Chinese word segmentation and part-of-speech tagging. In *Proceedings of Coling*, pp. 385–392, 2008.

[8] Nobuhiro Kaji and Masaru Kitsuregawa. Efficient word lattice generation for joint word segmentation and POS tagging in Japanese. In *Proceedings of IJCNLP*, pp. 153–161, 2013.

[9] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of EMNLP*, pp. 230–237, 2004.

[10] Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus while improving the parsing system. In *Proceedings of LREC*, pp. 719–724, 1998.

[11] Graham Neubig, Yousuke Nakata, and Shinsuke Mori. Pointwise prediction for robust adaptable Japanese morphological analysis. In *Proceedings of ACL*, pp. 529–533, 2011.

[12] Ryohei Sasano, Sadao Kurohashi, and Manabu Okumura. A simple approach to unknown word processing in Japanese morphological analysis. In *Proceedings of IJCNLP*, pp. 162–170, 2013.

[13] Xu Sun, Takuya Matsuzaki, and Wenjie Li. Latent structured perceptrons for large-scale learning with hidden information. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 9, pp. 2063–2075, 2013.

[14] Pidong Wang and Hwee Tou Ng. A beam-search decoder for normalization of social media text with application to machine translation. In *Proceedings of NAACL*, pp. 471–481, 2013.

[15] Yunqing Xia, Kam-Fai Wong, and Wenjie Li. A phonetic-based approach to Chinese chat text normalization. In *Proceedings of ACL*, pp. 993–1000, 2006.

[16] Yi Yang and Jacob Eisenstein. A log-linear model for unsupervised text normalization. In *Proceedings of EMNLP*, pp. 61–72, 2013.

[17] 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. 言語処理学会年次大会論文集, pp. 1272–1275, 2012.

[18] 鍛治伸裕, 吉永直樹, 喜連川優. マイクロブログに対する形態素・正規化情報のアノテーション. 言語処理学会年次大会論文集, 2014.