Accurate Word Segmentation and POS Tagging for Japanese Microblogs: Corpus Annotation and Joint Modeling with Lexical Normalization

Nobuhiro Kaji[*†] and **Masaru Kitsuregawa**[†‡]
[*]National Institute of Information and Communications Technology
[†]Institute of Industrial Science, The University of Tokyo
[‡]National Institute of Informatics
{kaji, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract

Microblogs have recently received widespread interest from NLP researchers. However, current tools for Japanese word segmentation and POS tagging still perform poorly on microblog texts. We developed an annotated corpus and proposed a joint model for overcoming this situation. Our annotated corpus of microblog texts enables not only training of accurate statistical models but also quantitative evaluation of their performance. Our joint model with lexical normalization handles the orthographic diversity of microblog texts. We conducted an experiment to demonstrate that the corpus and model substantially contribute to boosting accuracy.

1 Introduction

Microblogs, such as Twitter[1] and Weibo[2], have recently become an important target of NLP technology. Since microblogs offer an instant way of posting textual messages, they have been given increasing attention as valuable sources for such actions as mining opinions (Jiang et al., 2011) and detecting sudden events such as earthquake (Sakaki et al., 2010).

However, many studies have reported that current NLP tools do not perform well on microblog texts (Foster et al., 2011; Gimpel et al., 2011). In the case of Japanese text processing, the most serious problem is poor accuracy of word segmentation and POS tagging. Since these two tasks are positioned as the fundamental step in the text processing pipeline, their accuracy is vital for all downstream applications.

[1]https://twitter.com
[2]https://www.weibo.com

1.1 Development of annotated corpus

The main obstacle that makes word segmentation and POS tagging in the microblog domain challenging is the lack of annotated corpora. Because current annotated corpora are from other domains, such as news articles, it is difficult to train models that perform well on microblog texts. Moreover, system performance cannot be evaluated quantitatively.

We remedied this situation by developing an annotated corpus of Japanese microblogs. We collected 1831 sentences from Twitter and manually annotated these sentences with word boundaries, POS tags, and normalized forms of words (*c.f.*, Section 1.2).

We, for the first time, present a comprehensive empirical study of Japanese word segmentation and POS tagging on microblog texts by using this corpus. Specifically, we investigated how well current models trained on existing corpora perform in the microblog domain. We also explored performance gains achieved by using our corpus for training, and by jointly performing lexical normalization (*c.f.*, Section 1.2).

1.2 Joint modeling with lexical normalization

Orthographic diversity in microblog texts causes a problem when training a statistical model for word segmentation and POS tagging. Microblog texts frequently contain informal words that are spelled in a non-standard manner, *e.g.*, "*oredi* (*already*)", "*b4* (*before*)", and "*talkin* (*talking*)" (Han and Baldwin, 2011). Such words, hereafter referred to as *ill-spelled words*, are so productive that they considerably increase the vocabulary size. This makes training of statistical models difficult.

We address this problem by jointly conducting lexical normalization. Although a wide variety of ill-spelled words are used in microblog texts, many can be normalized into *well-spelled equivalents*, which conform to standard rules of spelling.

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 99–109,
October 25-29, 2014, Doha, Qatar.

A joint model with lexical normalization is able to handle orthographic diversity by exploiting information obtainable from the well-spelled equivalents.

The proposed joint model was empirically evaluated on the microblog corpus we developed. Our experiment demonstrated that the proposed model can perform word segmentation and POS tagging substantially better than current state-of-the-art models.

1.3 Summary

Contributions of this paper are the following:

- We developed a microblog corpus that enables not only training of accurate models but also quantitative evaluation for word segmentation and POS tagging in the microblog domain.[3]
- We propose a joint model with lexical normalization for better handling of orthographic diversity in microblog texts. In particular, we present a new method of training the joint model using a partially annotated corpus (*c.f.*, Section 7.4).
- We, for the first time, present a comprehensive empirical study of word segmentation and POS tagging for microblogs. The experimental results demonstrated that both the microblog corpus and joint model greatly contributes to training accurate models for word segmentation and POS tagging.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 discusses the task of lexical normalization and introduces terminology. Section 4 presents our microblog corpus and results of our corpus analysis. Section 5 presents an overview of our joint model with lexical normalization, and Sections 6 and 7 provide details of the model. Section 8 presents experimental results and discussions, and Section 9 presents concluding remarks.

2 Related Work

Researchers have recently developed various microblog corpora annotated with rich linguistic information. Gimpel et al. (2011) and Foster et al. (2011) annotated English microblog posts with POS tags. Han and Baldwin (2011) released a microblog corpus annotated with normalized forms of words. A Chinese microblog corpus annotated with word boundaries was developed for SIGHAN bakeoff (Duan et al., 2012). However, there are no microblog corpora annotated with word boundaries, POS tags, and normalized sentences.

There has been a surge of interest in lexical normalization with the advent of microblogs (Han and Baldwin, 2011; Liu et al., 2012; Han et al., 2012; Wang and Ng, 2013; Zhang et al., 2013; Ling et al., 2013; Yang and Eisenstein, 2013; Wang et al., 2013). However, these studies did not address enhancing word segmentation.

Wang et al. (2013) proposed a method of joint ill-spelled word recognition and word segmentation. With their method, informal spellings are merely recognized and not normalized. Therefore, they did not investigate how to exploit the information obtainable from well-spelled equivalents to increase word segmentation accuracy.

Some studies also explored integrating the lexical normalization process into word segmentation and POS tagging (Ikeda et al., 2009; Sasano et al., 2013). A strength of our joint model is that it uses rich character-level and word-level features used in state-of-the-art models of joint word segmentation and POS tagging (Kudo et al., 2004; Neubig et al., 2011; Kaji and Kitsuregawa, 2013). Thanks to these features, our model performed much better than Sasano et al.'s system, which is the only publicly available system that jointly conducts lexical normalization, in the experiments (see Section 8). Another advantage is that our model can be trained on a partially annotated corpus. Furthermore, we present a comprehensive evaluation in terms of precision and recall on our microblog corpus. Such an evaluation has not been conducted in previous work due to the lack of annotated corpora.[4]

3 Lexical Normalization Task

This section explains the task of lexical normalization addressed in this paper. Since lexical normalization is a relatively new research topic, there are no precise definitions of a lexical normalization task that are widely accepted by researchers.

[3] Please contact the first author for this corpus.

[4] Very recently, Saito et al. (2014) conducted similar empirical evaluation on microblog corpus. However, they used biased dataset, in which every sentence includes at least one ill-spelled words.

Table 1: Examples of our target ill-spelled words and their well-spelled equivalents. Phonemes are shown between slashes. English translations are provided in parentheses.

Ill-spelled word	Well-spelled equivalent
すげえ /sugee/	すごい /sugoi/ (great)
戻ろ /modoro/	戻ろう /modorou/ (going to return)
うまいいいい /umaiiii/	うまい /umai/ (yummy)

Therefore, it is important to clarify our task setting before discussing our joint model.

3.1 Target ill-spelled words

Many studies on lexical normalization have pointed out that phonological factors are deeply involved in the process of deriving ill-spelled words. Xia et al. (2006) investigated a Chinese chat corpus and reported that 99.2% of the ill-spelled words were derived by phonetic mapping from well-spelled equivalents. Wang and Ng (2013) analyzed 200 Chinese messages from Weibo and 200 English SMS messages from the NUS SMS corpus (How and Kan, 2005). Their analysis revealed that most ill-spelled words were derived from well-spelled equivalents based on pronunciation similarity.

On top of these investigations, we focused on ill-spelled words that are derived by phonological mapping from well-spelled words by assuming that such ill-spelled words are dominant in Japanese microblogs as well. We also assume that these ill-spelled words can be normalized into well-spelled equivalents on a word-to-word basis, as assumed in a previous study (Han and Baldwin, 2011). The validity of these two assumptions is empirically assessed in Section 4.

Table 1 lists examples of our target ill-spelled words, their well-spelled equivalents, and their phonemes. The ill-spelled word in the first row is formed by changing the continuous two vowels from /oi/ to /ee/. This type of change in pronunciation is often observed in Japanese spoken language. The second row presents contractions. The last vowel character "う" /u/ of the well-spelled word is dropped. The third row illustrates word lengthening. The ill-spelled word is derived by repeating the vowel character "い" /i/.

3.2 Terminology

We now introduce the terminology that will be used throughout the remainder of this paper. The term *word surface form* (or *surface form* for short) is used to refer to the word form observed in an actual text, while *word normal form* (or *normal form*) refers to the normalized word form. Note that surface forms of well-spelled words are always identical to their normal forms.

It is possible that the word surface form and normal form have distinct POS tags, although they are identical in most cases. Take the ill-spelled word "戻ろ" /modoro/ as an example (the second row of Table 1). According to the JUMAN POS tag set,[5] POS of its surface form is CONTRACTED VERB, while that of its normal form is VERB.[6] To handle such a case, we strictly distinguish between these two POS tags by referring to them as *surface POS tags* and *normal POS tags*, respectively.

Given these terms, the tasks addressed in this paper can be stated as follows. Word segmentation is a task of segmenting a sentence into a sequence of word surface forms, and POS tagging is a task of providing surface POS tags. The task of joint lexical normalization, word segmentation, and POS tagging is to map a sentence into a sequence of quadruplets: word surface form, surface POS tag, normal form, and normal POS tag.

4 Microblog Corpus

This section introduces our microblog corpus. We first explain the process of developing the corpus then present the results of our agreement study and corpus analysis.

4.1 Data collection and annotation

The corpus was developed by manually annotating text messages posted to Twitter.

The posts to be annotated were collected as follows. 171,386 Japanese posts were collected using the Twitter API[7] on December 6, 2013. Among these, 1000 posts were randomly selected then manually split into sentences. As a result, we obtained 1831 sentences as a source of the corpus.

Two human participants annotated the 1831 sentences with surface forms and surface POS tags. Since much effort has already been done to annotate corpora with this information, the annotation process here follows the guidelines used to

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

[6] In this paper, we use simplified POS tags for explanation purposes. Remind that these tags are different from the original ones defined in JUMAN POS tag set.

[7] https://stream.twitter.com/1.1/statuses/sample.json

develop such corpora in previous studies (Kurohashi and Nagao, 1998; Hashimoto et al., 2011).

The two participants also annotated ill-spelled words with their normal forms and normal POS tags. Although this paper targets only informal phonological variations (*c.f.*, Section 3), other types of ill-spelled words were also annotated to investigate their frequency distribution in microblog texts. Specifically, besides informal phonological variations, spelling errors and Twitter-specific abbreviations were annotated. As a result, 833 ill-spelled words were identified (Table 2). They were all annotated with normal forms and normal POS tags.

4.2 Agreement study

We investigated the inter-annotator agreement to check the reliability of the annotation. During the annotation process, the two participants collaboratively annotated around 90% of the sentences (specifically, 1647 sentences) with normal forms and normal POS tags, and elaborated an annotation guideline through discussion. They then independently annotated the remaining 184 sentences (1431 words), which were used for the agreement study. Our annotation guideline is shown in the supplementary material.

We first explored the extent to which the two participants agreed in distinguishing between well-spelled words and ill-spelled words. For this task, we observed Cohen's kappa of 0.96 (almost perfect agreement). This results show that it is easy for humans to distinguish between these two types of words.

Next, we investigated whether the two participants could give ill-spelled words with the same normal forms and normal POS tags. For this purpose, we regarded the normal forms and normal POS tags annotated by one participant as goldstandards and calculated precision and recall achieved by the other participant. We observed moderate agreement between the two participants: 70% (56/80) precision and 73% (56/76) recall. We manually analyzed the conflicted examples and found that there were more than one acceptable normal form in many of these cases. Therefore, we would like to note that the precision and recall reported above are rather pessimistic estimations.

4.3 Analysis

We conducted corpus analysis to confirm the feasibility of our approach.

Table 2: Frequency distribution over three types of ill-spelled words in corpus.

Type	Frequency
Informal phonological variation	804 (92.9%)
Spelling error	27 (3.1%)
Twitter-specific abbreviation	34 (3.9%)
Total	865 (100%)

Table 2 illustrates that phonological variations constitute a vast majority of ill-spelled words in Japanese microblog texts. In addition, analysis of the 804 phonological variations showed that 793 of them can be normalized into single words. These represent the validity of the two assumptions we made in Section 3.1.

We then investigated whether lexical normalization can decrease the number of out-of-vocabulary words. For the 793 ill-spelled words, we counted how many of their surface forms and normal forms were not registered in the JUMAN dictionary.[8] The result suggests that 411 (51.8%) and 74 (9.3%) are not registered in the dictionary. This indicates the effectiveness of lexical normalization for decreasing out-of-vocabulary words.

5 Overview of Joint Model

This section gives an overview of our joint model with lexical normalization for accurate word segmentation and POS tagging.

5.1 Lattice-based approach

A lattice-based approach has been commonly adopted to perform joint word segmentation and POS tagging (Jiang et al., 2008; Kudo et al., 2004; Kaji and Kitsuregawa, 2013). In this approach, an input sentence is transformed into a word lattice in which the edges are labeled with surface POS tags (Figure 1). Given such a lattice, word segmentation and POS tagging can be performed at the same time by traversing the lattice. A discriminative model is typically used for the traversal.

An advantage of this approach is that, while the lattice can represent an exponentially large number of candidate analyses, it can be quickly traversed using dynamic programming (Kudo et al., 2004; Kaji and Kitsuregawa, 2013) or beam search (Jiang et al., 2008). In addition, a discriminative model allows the use of rich word-level features to find the correct analysis.

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

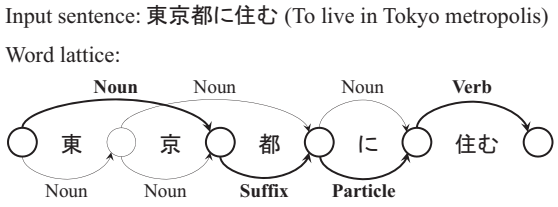

Figure 1: Example lattice (Kudo et al., 2004; Kaji and Kitsuregawa, 2013). Circle and arrow represent node and edge, respectively. Bold edges represent correct analysis.

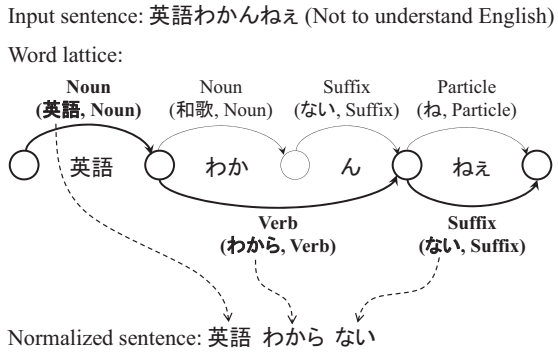

Figure 2: Lattice used to perform joint task. Normal forms and normal POS tags are shown in parentheses. As indicated by dotted arrows, normalized sentence can be obtained by concatenating normal forms associated with edges in correct analysis.

We propose extending the lattice-based approach to jointly perform lexical normalization, word segmentation, and POS tagging. We transform an input sentence into a word lattice in which the edges are labeled with not only surface POS tags but normal forms and normal POS tags (Figure 2). By traversing such a lattice, the three tasks can be performed at the same time. This approach can not only exploit rich information obtainable from word normal forms, but also achieve efficiency similar to the original lattice-based approach.

5.2 Issues

Issues on how to develop this lattice-based approach is detailed in Sections 6 and 7.

Section 6 describes how to generate a word lattice from an input sentence. This is done using a hybrid approach that combines a statistical model and *normalization dictionary*. The normalization dictionary is specifically a list of quadruplets: word surface form, surface POS tag, normal form, and normal POS tag (Table 3).

Table 3: Normalization dictionary. Columns represent entry ID, surface form, surface POS, normal form, and normal POS, respectively.

ID	Surf.	Surf. POS	Norm.	Norm. POS
A	すごい	ADJECTIVE	すごい	ADJECTIVE
B	すげえ	ADJECTIVE	すごい	ADJECTIVE
C	戻ろう	VERB	戻ろう	VERB
D	戻ろ	CONTR. VERB	戻ろう	VERB
E	うまい	ADJECTIVE	うまい	ADJECTIVE
F	うまいいいい	ADJECTIVE	うまい	ADJECTIVE

Table 4: Tag dictionary.

ID	Surf. form	Surf. POS
a	すごい (great)	ADJECTIVE
b	戻ろう (going to return)	VERB
c	戻ろ (gonna return)	CONTR. VERB
d	うまい (yummy)	ADJECTIVE

Section 7 describes a discriminative model for the lattice traversal. Our feature design as well as two training methods are presented.

6 Word Lattice Generation

In this section, we first describe a method of constructing a normalization dictionary then present a method of generating a word lattice from an input sentence.

6.1 Construction of normalization dictionary

Although large-scale normalization dictionaries are difficult to obtain, *tag dictionaries*, which list pairs of word surface forms and their surface POS tags (Table 4), are widely available in many languages including Japanese. Therefore, we use an existing tag dictionary to construct the normalization dictionary.

Due to space limitations, we give only a brief overview of our construction method, omitting its details. We note that our method uses hand-crafted rules similar to those used in (Sasano et al., 2013); hence, the proposal of this method is not an important contribution. To make our experimental results reproducible, our normalization dictionary, as well as a tool for constructing it, is released as supplementary material.

Our method of constructing the normalization dictionary takes three steps. The following explains each step using Tables 3 and 4 as running examples.

Step 1 A tag dictionary generally contains a small number of ill-spelled words, although well-spelled words constitute a vast majority. We identify such ill-spelled words by using a manually-tailored list of surface POS tags indicative of informal spelling (*e.g.*, CONTRACTED VERB). For example, entry (c) in Table 4 is identified as an ill-spelled word in this step.

Step 2 The tag dictionary is augmented with normal forms and normal POS tags to construct a small normalization dictionary. For ill-spelled words identified in step 1, the normal forms and normal POS tags are determined by hand-crafted rules. For example, the normal form is derived by appending the vowel character "う" /u/ to the surface form, if the surface POS tag is CONTRACTED VERB. This rule derives entry (D) in Table 3 from entry (c) in Table 4. For well-spelled words, on the other hand, the normal forms and normal POS tags are simply set the same as the surface forms and surface POS tags. For example, entries (A), (C), and (E) in Table 3 are generated from entries (a), (b), and (d) in Table 4, respectively.

Step 3 Because the normalization dictionary constructed in step 2 contains only a few ill-spelled words, it is expanded in this step. For this purpose, we use hand-crafted rules to derive ill-spelled words from the entries already registered in the normalization dictionary. Some rules are taken from (Sasano et al., 2013), while the others are newly tailored. In Table 3, for example, entry (B) is derived from entry (A) by applying the rule that substitutes "ごい" /goi/ with "げえ" /gee/.

A small problem that arises in step 3 is how to handle lengthened words, such as entry (F) in Table 3. While lengthened words can be easily derived using simple rules (Brody and Diakopoulos, 2011; Sasano et al., 2013), such rules infinitely increase the number of entries because an unlimited number of lengthened words can be derived by repeating characters. To address this problem, no lengthened words are added to the normalization dictionary in step 3. We instead use rules to skip repetitive characters in an input sentence when performing dictionary match.

6.2 A hybrid approach

A word lattice is generated using both a statistical method (Kaji and Kitsuregawa, 2013) and the normalization dictionary.

We begin by generating a word lattice which encodes only word surface forms and surface POS tags (*c.f.*, Figure 1) using the statistical method proposed by Kaji and Kitsuregawa (2013). Interested readers may refer to their paper for details.

Each edge in the lattice is then labeled with normal forms and normal POS tags. Note that a single edge can have more than one candidate normal form and normal POS tag. In such a case, new edges are accordingly added to the lattice.

The edges are labeled with normal forms and normal POS tags in the following manner. First, every edge is labeled with a normal form and normal POS tag that are identical with the surface form and surface POS tag. This is based on our observation that most words are well-spelled ones. The edge is not provided with further normal forms and normal POS tags, if the normalization dictionary contains a well-spelled word that has the same surface form as the edge. Otherwise, we allow the edge to have all pairs of normal forms and normal POS tags that are obtained by using the normalization dictionary.

7 Discriminative Lattice Traversal

This section explains a discriminative model for traversing the word lattice. The lattice traversal with a discriminative model can formally be written as

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

Here, x denotes an input sentence, $\boldsymbol{w}$, $\boldsymbol{t}$, $\boldsymbol{v}$, and $\boldsymbol{s}$ denote a sequence of word surface forms, surface POS tags, normal forms, and normal POS tags, respectively, $\mathcal{L}(x)$ represents a set of candidate analyses represented by the word lattice, and $\boldsymbol{f}(\cdot)$ and $\boldsymbol{\theta}$ are feature and weight vectors.

We now describe features, a decoding method, and two training methods.

7.1 Features

We use character-level and word-level features used for word segmentation and POS tagging in (Kaji and Kitsuregawa, 2013). To take advantage of joint model with lexical normalization, the word-level features are extracted from not only surface forms but also normal forms. See (Kaji and Kitsuregawa, 2013) for the original features.

In addition, several new features are introduced in this paper. We use the quadruplets (w_i, t_i, v_i, s_i)

and pairs of surface and normal POS tags (t_i, s_i) as binary features to capture probable mappings between ill-spelled words and their well-spelled equivalents. We use another binary feature indicating whether a quadruplet (w_i, t_i, v_i, s_i) is registered in the normalization dictionary. Also, we use a bigram language model feature, which prevents sentences from being normalized into ungrammatical and/or incomprehensible ones. The language model features are associated with normalized bigrams, $(v_{i-1}, s_{i-1}, v_i, s_i)$, and take as the values the logarithmic frequency $\log_{10}(f+1)$, where f represents the bigram frequency (Kaji and Kitsuregawa, 2011). Since it is difficult to obtain a precise value of f, it is approximated by the frequency of the surface bigram, $(w_{i-1}, t_{i-1}, w_i, t_i)$, calculated from a large raw corpus automatically analyzed using a system of joint word segmentation and POS tagging. See Section 8.1 for the raw corpus and system used in the experiments.

7.2 Decoding

It is easy to find the best analysis $(\boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s})$ among the candidates represented by the word lattice. Although we use several new features, we can still locate the best analysis by using the same dynamic programming algorithm as in previous studies (Kudo et al., 2004; Kaji and Kitsuregawa, 2013).

7.3 Training on a fully annotated corpus

It is straightforward to train the joint model provided with a fully annotated corpus, which is labeled with word surface forms, surface POS tags, normal forms, and normal POS tags.

We use structured perceptron (Collins, 2002) for the training (Algorithm 1). The training begins by initializing $\boldsymbol{\theta}$ as a zero vector (line 1). It then reads the annotated corpus $\mathcal{C}$ (line 2-9). Given a training example, $(x, \boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s}) \in \mathcal{C}$, the algorithm locates the best analysis, $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$, based on the current weight vector (line 4). If the best analysis differs from the oracle analysis, $(\boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s})$, the weight vector is updated (line 5-7). After going through the annotated corpus m times (m=10 in our experiment), the averaged weight vector is returned (line 10).

7.4 Training on a partially annotated corpus

Although the training with the perceptron algorithm requires a fully annotated corpus, it is labor-intensive to fully annotate sentences. This consideration motivates us to explore training our model with less supervision. We specifically explore using a corpus annotated with only word boundaries and POS tags.

Algorithm 1 Perceptron training

1: $\boldsymbol{\theta} \leftarrow \mathbf{0}$
2: **for** $i = 1 \ldots m$ **do**
3: **for** $(x, \boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s}) \in \mathcal{C}$ **do**
4: $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \leftarrow \textsc{Decoding}(x, \boldsymbol{\theta})$
5: **if** $(\boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s}) \neq (\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$ **then**
6: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{f}(x, \boldsymbol{w}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s}) - \boldsymbol{f}(x, \hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$
7: **end if**
8: **end for**
9: **end for**
10: **return** $\textsc{Average}(\boldsymbol{\theta})$

Algorithm 2 Latent perceptron training

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: $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \leftarrow \textsc{Decoding}(x, \boldsymbol{\theta})$
5: $(\boldsymbol{w}, \boldsymbol{t}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{s}}) \leftarrow \textsc{ConstrainedDecoding}(x, \boldsymbol{\theta})$
6: **if** $\boldsymbol{w} \neq \hat{\boldsymbol{w}}$ or $\boldsymbol{t} \neq \hat{\boldsymbol{t}}$ **then**
7: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{f}(x, \boldsymbol{w}, \boldsymbol{t}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{s}}) - \boldsymbol{f}(x, \hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$
8: **end if**
9: **end for**
10: **end for**
11: **return** $\textsc{Average}(\boldsymbol{\theta})$

We use the latent perceptron algorithm (Sun et al., 2013) to train the joint model from such a partially annotated corpus (Algorithm 2). In this scenario, a training example is a sentence x paired with a sequence of word surface forms $\boldsymbol{w}$ and surface POS tags $\boldsymbol{t}$ (*c.f.*, line 3). Similarly to the perceptron algorithm, we locate the best analysis $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{s}})$ for a given training example, (line 4). We also locate the best analysis, $(\boldsymbol{w}, \boldsymbol{t}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{s}})$, among those having the same surface forms $\boldsymbol{w}$ and surface POS tags $\boldsymbol{t}$ as the training example (line 5). If the surface forms and surface POS tags of the former analysis differ from the annotations of the training example, parameter is updated by regarding the latter analysis as an oracle (line 6-8).

8 Experiments

We conducted experiments to investigate how the microblog corpus and joint model contribute to improving accuracy of word segmentation and POS tagging in the microblog domain.

8.1 Setting

We constructed the normalization dictionary from the JUMAN dictionary 7.0.[9] While JUMAN dic-

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

tionary contains 750,156 entries, the normalization dictionary contains 112,458,326 entries.

Some features taken from the previous study (Kaji and Kitsuregawa, 2013) are induced using a tag dictionary. For this we used two tag dictionaries. One is JUMAN dictionary 7.0 and the other is a tag dictionary constructed by listing surface forms and surface POS tags in the normalization dictionary.

To compute the language model features, one billion sentences from Twitter posts were analyzed using MeCab 0.996.[10] We used all bigrams appearing at least 10 times in the auto-analyzed sentences.

8.2 Results of word segmentation and POS tagging

We first investigated the performance of models trained on an existing annotated corpus form news texts. For this experiment, our joint model as well as three state-of-the-art models (Kudo et al., 2004)[11](Neubig et al., 2011)[12](Kaji and Kitsuregawa, 2013) were trained on Kyoto University Text corpus 4.0 (Kurohashi and Nagao, 1998). Since this training corpus is not annotated with normal forms and normal POS tags, our model was trained using the latent perceptron. Table 5 summarizes the word-level F_1-scores (Kudo et al., 2004) on our microblog corpus. The two columns represent the results for word segmentation (**Seg**) and joint word segmentation and POS tagging (**Seg+Tag**), respectively.

We also conducted 5-fold crossvalidation on our microblog corpus to evaluate performance improvement when these models are trained on microblog texts (Table 6). In addition to the models in Table 5, results of a rule-based system (Sasano et al., 2013)[13] and our joint model trained using the perceptron algorithm are also presented. Notice that **Proposed** and **Proposed (latent)** represent our model trained using perceptron and latent perceptron, respectively.

From Tables 5 and 6, as expected, we see that the models trained on news texts performed poorly on microblog texts, while their performance significantly boosted when trained on the microblog texts. This demonstrates the importance of corpus annotation. An exception was **Kudo04**. Its performance improved only slightly, even when it was trained on the microblog texts. We believe this is because their model uses dictionary-based rules to prune candidate analyses; thus, it could not perform well in the microblog domain, where out-of-vocabulary words are abundant.

Table 5: Performance of models trained on the news articles.

	Seg	Seg+Tag
Kudo04	81.8	71.0
Neubig11	80.5	69.1
Kaji13	**83.2**	73.1
Proposed (latent)	83.0	**73.9**

Table 6 also illustrates that our joint models achieved F_1-score better than the state-of-the-art models trained on the microblog texts. This shows that modeling the derivation process of ill-spelled words makes training easier. We conducted bootstrap resampling (with 1000 samples) to investigate the significance of the improvements achieved with our joint model. The results showed that all improvements over the baselines were statistically significant ($p < 0.01$). The difference between **Proposed** and **Proposed (latent)** were also statistically significant ($p < 0.01$).

The results of **Proposed (latent)** are interesting. Table 5 illustrates that our joint model performs well even when it is trained on a news corpus that rarely contains ill-spelled words and is not at all annotated with normal forms and normal POS tags. This indicates the robustness of our training method and the importance of modeling word derivation process in the microblog domain. In Table 6, we observed that **Proposed (latent)**, which uses less supervision, performed better than **Proposed**. The reason for this will be examined later.

In summary, we can conclude that both the microblog corpus and joint model significantly contribute to training accurate models for word segmentation and POS tagging in the microblog domain.

8.3 Results of lexical normalization

While the main goal with this study was to enhance word segmentation and POS tagging in the microblog domain, it is interesting to explore how well our joint model can normalize ill-spelled words.

Table 7 illustrates precision, recall, and F_1-score for the lexical normalization task. To put

[10] https://code.google.com/p/mecab
[11] https://code.google.com/p/mecab
[12] http://www.phontron.com/kytea/
[13] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

Table 6: Results of 5-fold cross-validation on microblog corpus.

	Seg	Seg+Tag
Kudo04	82.7	71.7
Neubig11	88.6	75.9
Kaji13	90.9	82.1
Sasano13	82.7	73.3
Proposed	91.3	83.2
Proposed (latent)	**91.4**	**83.7**

Table 7: Results of lexical normalization task in terms of precision, recall, and F_1-score.

	Precision	Recall	F_1
Neubig11	69.2	35.9	47.3
Proposed	**77.1**	**44.6**	**56.6**
Proposed (latent)	53.7	24.7	33.9

the results into context, we report on the baseline results of a tagging model proposed by Neubig et al. (2011). This baseline conducts lexical normalization by regarding it as two independent tagging tasks (*i.e.*, tasks of tagging normal forms and normal POS tags). The result of the baseline model is also obtained using 5-fold crossvalidation.

Table 7 illustrates that **Proposed** performed significantly better than the simple tagging model, **Neubig11**. This suggests the effectiveness of our joint model. On the other hand, **Proposed (latent)** performed poorly in this task. From this result, we can argue that **Proposed (latent)** can achieve superior performance in word segmentation and POS tagging (Table 6) because it gave up correctly normalizing ill-spelled words, focusing on word segmentation and POS tagging.

The experimental results so far suggest the following strategy for training our joint model. If accuracy of word segmentation and POS tagging is the main concern, we can use the latent perceptron. This approach has the advantage of being able to use a partially annotated corpus. On the other hand, if performance of lexical normalization is crucial, we have to use the standard perceptron algorithm.

8.4 Error analysis

We manually analyzed erroneous outputs and observed several tendencies.

We found that a word lattice sometimes missed the correct output. Such an error was, for example, observed in a sentence including many ill-spelled words, e.g., '周囲の目が、キニナリマス！ (be nervous about what other people think!)', where the part 'キニナリマス' is in ill-spelled words. Improving the lattice generation algorithm is considered necessary to achieve further performance gain.

Even if the correct analysis appears in the word lattice, our model sometimes failed to handle ill-spelled words, incorrectly analyzing them as out-of-vocabulary words. For example, the proposed method treated the phrase 'おやつたーいむ (snack time)' as a single out-of-vocabulary word, even though the correct analysis was found in the word lattice. More sophisticated features would be required to accurately distinguish between ill-spelled and out-of-vocabulary words.

9 Conclusion and Future Work

We presented our attempts towards developing an accurate model for word segmentation and POS tagging in the microblog domain. To this end, we, for the first time, developed an annotated corpus of microblogs. We also proposed a joint model with lexical normalization to handle orthographic diversity in the microblog text. Intensive experiments demonstrated that we could successfully improve the performance of word segmentation and POS tagging on microblog texts. We believe this study will have a large practical impact on a various research areas that target microblogs.

One limitation of our approach is that it cannot handle certain types of ill-spelled words. For example, the current model cannot handle the cases in which there are no one-to-one-mappings between well-spelled and ill-spelled words. Also, our model cannot handle spelling errors, which are considered relatively frequent in the microblog than news domains. The treatment of these problems would require further research.

Another future research is to speed-up our model. Since the joint model with lexical normalization significantly increases the search space, it is much slower than the original lattice-based model for word segmentation and POS tagging.

Acknowledgments

The authors would like to thank Naoki Yoshinaga for his help in developing the microblog corpus as well as fruitful discussions.

References

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

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8.

Huiming Duan, Zhifang Sui, Ye Tian, and Wenjie Li. 2012. The CIPS-SIGHAN CLP 2012 Chinese word segmentation on microblog corpora bakeoff. In *Proceedings of the Second CIPS-SIGHAN Joint Conrerence on Chinese Language Processing*, pages 35–40.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. #hardtoparse: POS tagging and parsing the twitterverse. In *Proceedings of AAAI Workshop on Analysing Microtext*, pages 20–25.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of ACL*, pages 42–47.

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

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of EMNLP-CoNLL*, pages 421–432.

Chikara Hashimoto, Sadao Kurohashi, Daisuke Kawahara, Keiji Shinzato, and Masaaki Nagata. 2011. Construction of a blog corpus with syntactic, anaphoric, and semantic annotations (in Japanese). *Journal of Natural Language Processing*, 18(2):175–201.

Yijiu How and Min-Yen Kan. 2005. Optimizing predictive text entry for short message service on mobile phones. In *Proceedings of Human Computer Interfaces International*.

Kazushi Ikeda, Tadashi Yanagihara, Kazunori Matsumoto, and Yasuhiro Takishima. 2009. Unsupervised text normalization approach for morphological analysis of blog documents. In *Proceedings of Australasian Joint Conference on Advances in Artificial Intelligence*, pages 401–411.

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

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *Proceedings of ACL*, pages 151–160.

Nobuhiro Kaji and Masaru Kitsuregawa. 2011. Splitting noun compounds via monolingual and bilingual paraphrasing: A study on Japanese Katakana words. In *Proceedings of EMNLP*, pages 959–969.

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

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

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

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2013. Paraphrasing 4 microblog normalization. In *Proceedings of EMNLP*, pages 73–84.

Xiaohua Liu, Ming Zhou, Xiangyang Zhou, Zhongyang Fu, and Furu Wei. 2012. Joint inference of named entity recognition and normalization for tweets. In *Proceedings of ACL*, pages 526–535.

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

Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. 2014. Morphological analysis for Japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING*, pages 1773–1782.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquak shakes Twitter users: real-time event detection by social sensors. In *Proceedings of WWW*, pages 851–860.

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

Xu Sun, Takuya Matsuzaki, and Wenjie Li. 2013. Latent structured perceptrons for large-scale learning with hidden information. *IEEE Transactions on Knowledge and Data Engineering*, 25(9):2063–2075.

Aobo Wang and Min-Yen Kan. 2013. Mining informal language from Chinese microtext: Joint word recognition and segmentation. In *Proceedings of ACL*, pages 731–741.

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

Aobo Wang, Min-Yen Kan, Daniel Andrade, Takashi Onishi, and Kai Ishikawa. 2013. Chinese informal word normalization: an experimental study. In *Proceedings of IJCNLP*, pages 127–135.

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

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

Congle Zhang, Tyler Baldwin, Howard Ho, Benny Kimelfeld, and Yunyao Li. 2013. Adaptive parser-centric text normalization. In *Proceedings of ACL*, pages 1159–1168.