Automatically Extracting Variant-Normalization Pairs for Japanese Text Normalization

Itsumi Saito **Kyosuke Nishida** **Kugatsu Sadamitsu***
Kuniko Saito **Junji Tomita**
NTT Media Intelligence Laboratories
{saito.itsumi, nishida.kyosuke}@lab.ntt.co.jp
{saito.kuniko, tomita.junji}@lab.ntt.co.jp
k.sadamitsu.ic@future.co.jp

Abstract

Social media texts, such as tweets from Twitter, contain many types of non-standard tokens, and the number of normalization approaches for handling such noisy text has been increasing. We present a method for automatically extracting pairs of a variant word and its normal form from unsegmented text on the basis of a pair-wise similarity approach. We incorporated the acquired variant-normalization pairs into Japanese morphological analysis. The experimental results show that our method can extract widely covered variants from large Twitter data and improve the recall of normalization without degrading the overall accuracy of Japanese morphological analysis.

1 Introduction

Social media texts contain many non-standard tokens (lexical variants), e.g., by lengthening ("goooood" for "good") or abbreviating them ("tmrw" for "tomorrow"). Current language processing systems often fail to analyze such non-standard tokens, so normalizing them into standard tokens as a preprocess is promising for analyzing such noisy texts robustly (Cook and Stevenson, 2009; Han et al., 2012; Li and Liu, 2012, 2014). The normalization task mainly consists of two components. One is detecting variant words and generating normalization candidates. The other is constructing a word lattice from possible normalization candidates and decoding to select the best normalized word sequences. Early work on normalization focused on supervised approaches using labeled text, e.g., an approach based on a statistical machine translation (Aw et al., 2006; Pennell and Liu, 2011). However, social network service (SNS) text has a dynamic nature, and large SNS text is costly to annotate. Recent work has been focused on unsupervised approaches. For example, Han et al. (2012) proposed generating variant-normalization pairs automatically on the basis of distributional similarity and string similarity. Hassan and Menezes (2013) developed the approach by using a graph-based approach. Yang and Eisenstein (2013) introduced a highly accurate unsupervised normalization model. As just described, unsupervised methods have been developed for English normalization tasks.

Japanese SNS text also contains variant words, and several normalization methods have been proposed (Sasano et al., 2013; Kaji and Kitsuregawa, 2014; Saito et al., 2014). The basic framework of Japanese normalization is quite similar to that of English normalization. However, the problem is more complicated in Japanese normalization because Japanese words are not segmented using explicit delimiters, so we have to estimate word segmentation simultaneously in the decoding step. Variant words are also more difficult to extract automatically in Japanese than in explicitly segmented languages such as English. Unlike English normalization, the approaches for generating normalization candidates in Japanese are based on manually created rules or supervised training using annotated text. Japanese normalization contains problems to which the English unsupervised approach is simply applied. Although the English unsupervised approach assumes that there are explicit word segmentations, conventional analyzers often fail to segment non-standard words in Japanese. Therefore, to extract variants in an unsupervised fashion, we have to introduce an idea to generate correct word segmentation of variant words.

*Present affiliation: Future Architect,Inc.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 937–946,
Taipei, Taiwan, November 27 – December 1, 2017

Our idea for this problem is to use short sentences and phrases in SNS text. SNS text, like tweets from Twitter, contains many short sentences and phrases consisting of a single word or several words. For Example, "おっはよーん！ (*ohayon*, Good Morning)" is a variant form of "おはよう！(*ohayou*)," and "ちょーさみー (*cho samii*, It is very cold)" is a variant form of "超 (*cho*, very)/寒い (*samui*, cold)." Since these short sentences often contain variant words, they can be used as efficient cues for extracting a variant word. Our idea is to not extract a variant-normalization pair in one step. Instead, we present a two-step normalization approach. In the first step, we extract coarse candidates for variant-normalization pairs from unlabeled text, and in the second step, we incorporate the extracted pairs into Japanese morphological analysis and normalization. The appropriate normalization candidates are selected in the second step. We use training data for morphological analysis in the second step but do not use annotated data in the first step. Therefore, we can efficiently extract many types of variant-normalization pairs that appear in real text.

The contributions of this study are summarized as follows.

- We developed a new method for extracting pairs of variant and normal forms from tweets, which have no explicit delimiters, by focusing on short phrases and sentences in Twitter.
- We incorporated the variant-normalization pairs extracted by our method from tweets into a Japanese morphological analysis method and statistically significantly improved the accuracy for variant words without degrading the overall accuracy for Japanese morphological analysis.

2 Background

2.1 Japanese Morphological Analysis

As we mentioned above, we have to consider Japanese normalization tasks with Japanese morphological analysis. In this section, we describe the basic idea of Japanese morphological analysis. Japanese Morphological analysis can be interpreted as ranking while using a word lattice and scores of each path (Kaji and Kitsuregawa, 2013). There are two points to consider in the analysis procedure: how to generate the word lattice and how to formulate the score of each path. In Japanese morphological analysis, the dictionary-based approach has been widely used to generate word lattices (Kudo et al., 2004; Kaji and Kitsuregawa, 2013). To calculate the score of each path, two main scores are widely used: the score for a candidate word and the score for a pair of adjacent parts-of-speech (POSs). We can consider other various scores by using discriminative model (Kudo et al., 2004; Kaji and Kitsuregawa, 2013).

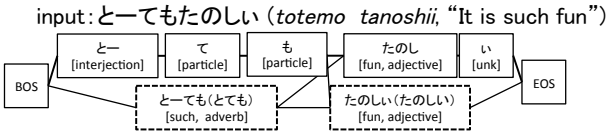

Figure 1: Example of Japanese morphological analysis and normalization

2.2 Related Work

Several studies have been conducted on Japanese morphological analysis and normalization. The approach proposed by Sasano et al. (2013) developed heuristics to flexibly search by using a simple, manually created derivational rule. Their system generates a normalized character sequence based on derivational rules and adds new nodes when generating the word lattice using dictionary lookup. Figure 1 presents an example of this approach. If the non-standard written sentence "とーてもたのしぃ (*totemo tanoshii*, It is such fun)" is input, the traditional dictionary-based system generates nodes that are described using solid lines, as shown in Figure 1. Since "とーても (*totemo*, such)" and "たのしぃ (*tanoshii*, fun)" are Out Of Vocabulary (OOVs), the traditional system cannot generate the correct word segments or POS tags. However, their system generates additional nodes for the OOVs, shown as broken line rectangles in Figure 1. In this case, derivational rules are used that substitute "ー" with "null" and "ぃ (i)" with "い (i)", and the system can generate the standard forms "とても (*totemo*, such)" and "たのしい (*tanoshii*, fun)" and their POS tags. If we can generate sufficiently appropriate rules, these approaches seem to be effective. However, there are many types of derivational patterns in SNS text, and they are difficult to all cover manually. Moreover, how to set the path score for appropriately

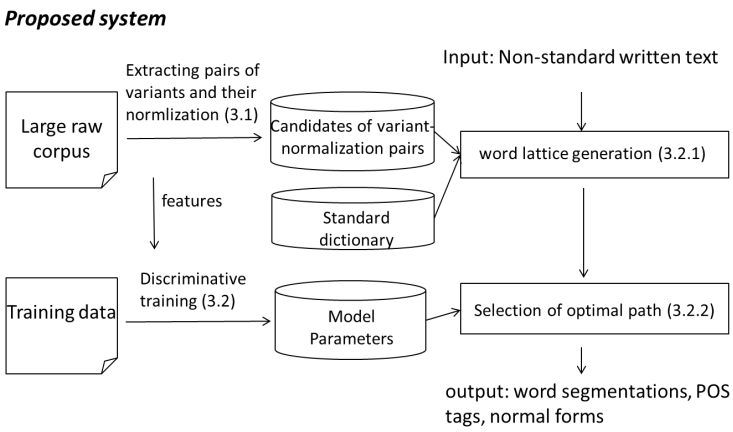

Figure 2: Overview of proposed system

ranking the word lattice when the number of candidates increases becomes a serious problem.

Saito et al. (2014) proposed supervised extraction of derivational patterns (we call them transformation patterns), incorporated these patterns into a word lattice, and formulated morphological analysis and normalization using a discriminate model. Although this approach can generate broad-coverage normalization candidates, it needs a large amount of annotation data of variant words and their normalization. Kaji and Kitsuregawa (2014) also proposed morphological analysis and normalization based on a discriminative model and created variant words on the basis of handmade rules. As far as we know, automatic extraction of variant-normalization pairs has not been researched. If we can extract variant-normalization pairs automatically, we can decrease the annotation cost and possibly increase accuracy by combining our method with other conventional methods.

Several studies have applied a character-based approach. For example, Sasaki et al. (2013) proposed a character-level sequential labeling method for normalization. However, it handles only one-to-one character transformations and does not take the word-level context into account. The proposed method can handle many-to-many character transformations and takes word-level context into account, so it has a wider scope for handling non-standard tokens.

Many studies have been done on text normalization for English; for example, Han and Baldwin (2011) classifies whether or not OOVs are non-standard tokens and estimates standard forms on the basis of contextual, string, and phonetic similarities. Han et al. (2012) and Hassan and Menezes (2013) developed the method of extracting variant-normalization pairs automatically for English. Yang and Eisenstein (2013) introduced a highly accurate unsupervised normalization model using log-linear model. In these studies, clear word segmentations were assumed to exist. However, since Japanese is unsegmented, the normalization problem needs to be treated as a joint normalization, word segmentation, and POS tagging problem.

Thus, we propose automatically extracting normalization candidates from unlabeled data and present a method for incorporating these candidates into Japanese morphological analysis and normalization. Our method can extract new variant patterns from real text.

3 Proposed Method

Our method consists of two parts. The first involves extracting normalization candidates and their normal forms from unlabeled data. The second involves a morphological analysis and normalization using extracted variants. Basically, we use a previously proposed dictionary based approach (Sasano et al., 2013; Saito et al., 2014; Kaji and Kitsuregawa, 2014), but the method for generating normalization candidates and some features used in a discriminative model are new. The proposed system is illustrated in Figure 2.

In the first part, we generate a coarsely segmented corpus and calculate the pairwise similarity of two arbitrary nodes that appear in the segmented corpus. In a previous study (Han and Baldwin, 2011), the nodes were assumed to be single words. On the other hand, our

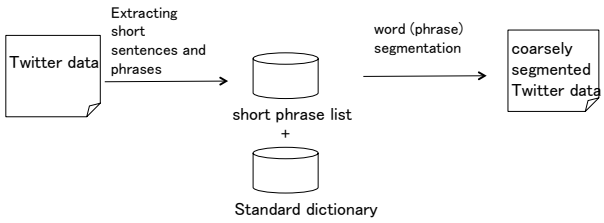

Figure 3: Flow of generating coarsely segmented corpus

method assigns a node to not only single words but also short phrases (See. 3.1). To calculate similarity between two nodes, we use semantic similarity and phonetic similarity. After calculating similarity, we filter the pairs that do not exceed a similarity threshold. We use word embeddings as a semantic similarity measure. We describe this more precisely in 3.1.2.

In the second part, the problem is how to incorporate extracted variants into Japanese morphological analysis. We use a discriminative model and Viterbi decoding for estimating word segmentation, POS tagging, and normalization. To prevent degradation induced by incorporating extracted variants, we introduce many types of features. We describe this more precisely in 3.2.

3.1 Extracting Candidates of Variant-Normalization Pairs from Twitter Texts

3.1.1 Preparation of Coarsely Segmented Corpus Using Short SNS Sentences and Phrases

We have to prepare a segmented corpus for generating normalization candidates and calculating similarity. The flow of generating coarsely segmented corpus is shown in Figure 3. As we mentioned above, we cannot determine the explicit word segmentation of unlabeled data, especially for variant words. However, we can assume that there are some explicit segmentations in text: the left and right ends of sentences and symbols such as punctuation, brackets, pictographs, emoticons, linefeed characters, commas, and spaces. SNS text contains many short sentences and phrases consisting of a word or several words. Our idea is to use the units of short sentences and short phrases delimited in symbols in SNS text as cues for extracting variant words.

More specifically, we first segment large Twitter text with several predefined symbols, extract character sequences consisting of ten or fewer characters, and insert them into a standard dictionary. Then we segment the large twitter corpus using an expanded dictionary and conventional analyzer. An example of a segmented sentence using an expanded dictionary is “おっはよーう (*ohhayou*)/ ! (Good morning!)”, whereas the result using a standard dictionary is “おっ(*o*)/は (*ha*)/よー (*yo*)/う (*u*)/ ! ” Since this segmented text contains several segmentation errors and noise, we extract reliable candidates using a similarity threshold described in the next subsection.

3.1.2 Similarity Measures

To calculate the similarity between two nodes w_i and w_j appearing in a segmented corpus (3.1.1), we mainly use two similarity measures: semantic and phonetic.

Semantic Similarity We calculate semantic similarity between w_i and w_j by inducing dense real-valued low-dimensional embeddings from large unlabeled text (Mikolov et al., 2013). We use the tool word2vec [1] to calculate embeddings of each node. Semantic similarity is defined as

$$semsim(w_i, w_j) = cos(vec(w_i), vec(w_j)) \quad (1)$$

This semantic similarity is used as a feature of a normalization and morphological analysis model (3.2.3). We set the embedding size is 200.

Phonetic Similarity We first convert a surface string into pronunciations (Japanese Kanji to Hiragana) and calculate the edit distance. We use two types of edit distance: standard and modified. For calculating modified edit distance, we set the substitution cost of two strings 0.5 when two strings have the same vowels, two strings have the same consonant or two strings are vowels. We also set insertion and deletion cost of vowels 0.5, while the standard cost of substitution, insertion and deletion is 1.

We use standard edit distance and modified edit distance as a threshold of candidate filtering (3.1.3) and modified edit distance as an element of a feature of a normalization and morphological analysis model (3.2.3). For a feature and threshold, we set the phonetic similarity $psim(w_i, w_j)$ as follows:

$$psim(w_i, w_j) = \prod_{m_i, m_j} p(m_i, m_j) \quad (2)$$

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

$$p(m_i, m_j) = 1 - MED(m_i, m_j)/OC(m_i, m_j) \quad (3)$$

Where $OC(m_i, m_j)$ indicates the total number of operations for calculating edit distance of m_i and m_j and $MED(m_i, m_j)$ indicates the modified edit distance. m_i and m_j indicate the morphemes in w_i and w_j, respectively.

To calculate morphological-level features, we analyzed w_j using conventional morphological analyzer and make morphological-level alignment using character-level alignment of w_i and w_j and morphological information of w_j. Here, w_j and w_i are regarded as a normal form and a variant form, respectively. Here is an example of w_i = "たんじょーび (birthday)" and w_j = "たんじょうび (birthday)". In this case, morphological-level alignment is (たんじょー/たんじょう, び/び) since character-level alignment is (た/た, ん/ん, じ/じ, ょ/ょ, ー/う, び/び) and word segmentation of w_j is (たんじょう, び).

3.1.3 Candidate Filtering using Similarity Measures

We calculate the pairwise similarity between two nodes appearing in a segmented corpus (3.1.1) and their filter using a similarity measure we defined (3.1.2). The set of nodes N consists of all tokens w that appear in the segmented corpus. Since there is a huge number of node pairs and most are irrelevant, we have to filter the pairs that have low similarities. Here, we set the threshold of $semsim$ to 0.4, and the threshold of standard phonetic edit distance is 2. Moreover, we filter the pairs in which the consonants of the beginning phonetic symbols of each morphemes are different or morphological-level phonetic similarity $p(m_i, m_j)$ is lower than 0.6. We also filtered the candidates when the number of consonants and all characters in a variant form (except for "っ", "ん", and lower case letters) are larger than that of a normal form at morphological level. If a variant word is already exists in a standard dictionary, we filtered the candidate. Note that this filtering is not intended to exactly identify whether the pair has the relationship of variant and normalization. Since we use only two similarities, simple phonetic information and the coarsely segmented corpus in this phase, we only extract candidates of variants and their normal forms coarsely in the first step. Then, we exactly identify the word segmentation and normalization simultaneously in the second step (3.2).

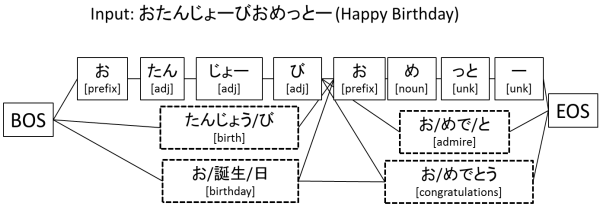

Figure 4: Example of proposed lattice

3.2 Normalization and Morphological Analysis

3.2.1 Incorporating Normalization Candidates in Japanese Morphological Analysis

We generate the word lattice using extracted candidate for variant-normalization pairs and dictionary lookup (See Figure 4). The broken line rectangles in Figure4 are nodes added by the proposed method. We exploit dictionary lookup by using the possible character sequence of the extracted normalized character sequences when variant character sequences match the input character sequences. For example, we exploit dictionary lookup for input character sequences such as "おたんじょーびおめっとー (happy birthday)" and add the possible normalized word sequences such as "お/誕生/日（birthday）" and "お/めでとう (congratulations)" which are from extracted variant-normalization candidates. The proposed method is intended to generate normalized word sequences. In the first step, appropriateness of word segmentation is not taken into account, but in this phase, we can exactly evaluate whether the acquired pair is appropriate for normalization or not by considering the morphological connectivity.

3.2.2 Objective Function

We used a discriminative model for incorporating many features. The decoder selects the optimal sequence $\hat{y}$ from $L(s)$ when given the candidate set $L(s)$ for sentence s. This is formulated as follows (Jiang et al., 2008; Kaji and Kitsuregawa, 2013):

$$\hat{y} = \arg\max_{y \in L(s)} \mathbf{w} \cdot \mathbf{f}(y) \quad (4)$$

Where $\hat{y}$ is the optimal path, $L(s)$ is the lattice created for s, $\mathbf{w} \cdot \mathbf{f}(y)$ is the dot product between

Name	Feature
Word unigram score	$f_{m_i p_i}$
POS bi-gram score	f_{p_{i-1},p_i}
Character sequence score	$1/\log(p'_{t_i}/p'_s)$ if $p'_{t_i} > p'_s$ otherwise 0. where, $p'_x = p_x^{1/length(x)}$, $p_x = \prod_{j=1}^{k} p(c_j \mid c_{j-5}^{j-1})$, $x \in (s, t_i)$
Character similarity score	$\log(psim)$
Semantic similarity score	$\log(semsim)$
Character frequency score	$\log(freq_w + 1)$
Character and morph transformation score	$\log(freq_{ct} + 1)$, $\log(freq_{mt} + 1)$, ϕ_{ct}, ϕ_{mt}

Table 1: Feature list

weight vector $\mathbf{w}$ and feature vector $\mathbf{f}(y)$. The optimal path is selected in accordance with the $\mathbf{w} \cdot \mathbf{f}(y)$ value. For estimating parameters, we used the averaged perceptron, which is widely used (Collins, 2002).

3.2.3 Features

The proposed lattice generation method generates a lattice larger than that generated in traditional dictionary-based lattice generation. Therefore, we need to introduce appropriate normalization scores into the objective function to prevent degradation. Table 1 lists the features we used. Let m_i be the ith word candidate and p_i be the POS tag of m_i. p_{i-1} and m_{i-1} are adjacent POS tag and word, respectively. We used the word unigram score $f_{m_i p_i}$, the cost for a pair of adjacent POSs f_{p_{i-1},p_i} that are estimated by MeCab [2], and additional scores.

The character sequence score reflects the character sequence probability of the normalization candidates (Saito et al., 2014). Here, s and t_i are input string and transformed string, respectively (e.g., in Figure 4, for the normalized node "お誕生日 (birthday)", s is "おたんじょーびおめっとー" and t_i is "お誕生日おめっとー". Then p_s and p_{t_i} are calculated by using the character 5-gram of a news corpus. c_j is the jth character of character sequence. We also used character sequence score as a candidate filter. We filtered the candidates that did not satisfy the pre-defined condition that $p'_s \leq p'_{t_i}$.

The character similarity score is calculated using $psim$ (see 3.1.2). The semantic similarity score is calculated using $semsim$ (see 3.1.2). The Character frequency score is a frequency of surface character sequences of variant nodes appeared in news data. Since variant words rarely appear in the news data, we use this feature to identify variant words and standard words. The character and morph transformation score is related to transformation patterns. $\log(freq_{ct} + 1)$ and $\log(freq_{mt} + 1)$ are the frequency of transformation patterns ct (character-level) and mt (morphological-level) that are extracted from variant-normalization candidates, respectively. ϕ_{ct} and ϕ_{mt} are 1 if a node contains transformation patterns ct and mt, otherwise 0, respectively. The scale of features were adjusted.

Since all those features can be factorized, the optimal path is searched by using the Viterbi algorithm.

3.2.4 Candidate Expansion

Although our method can extract many variants, we expand the variants to achieve higher recall. We use a simple rule for adding simple variation in the decoding step. For example, first, repetitions of more than one character of "ー", "〜" and "っ" are reduced to one character and repetitions of more than three characters of Japanese Hiragana and Katakana are reduced to three characters and one character. Moreover, we use the patterns of deletions of "ー", "〜", "っ" and lowercase characters (Saito et al., 2014).

4 Experiments

4.1 Data and Settings

We prepared the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa et al., 2014), which is a commonly used dataset in Japan and Twitter data. For unsupervised variant extraction, we used about 70 million unlabeled Twitter corpora. We used 2,000 sentences of BCCWJ text for training the decoder and Twitter data for the test. Twitter data contain manually annotated

[2] http://taku910.github.io/mecab/ (in Japanese)

variant forms	norm forms	translation	$semsim$
うれしーい (*ureshiii*)	うれしい (*ureshii*)	happy	0.655
うれしー (*ureshii*)	うれしい (*ureshii*)	happy	0.684
うれしぃ (*ureshii*)	うれしい (*ureshii*)	happy	0.575
うれすぃ (*uresui*)	うれしい (*ureshii*)	happy	0.568
うれちい (*uretii*)	うれしい (*ureshii*)	happy	0.649
うれしす (*ureshishu*)	うれしい (*ureshii*)	happy	0.715
かわええ (*kawaee*)	かわいい (*kawaii*)	cute	0.744
かんわいい (*kanwaii*)	かわいい (*kawaii*)	cute	0.683
きゃわいい (*kyawaii*)	かわいい (*kawaii*)	cute	0.770
お/ねげー/し/ます (*o/negee/shi/masu*)	お/ねがい/し/ます (*o/negai/shi/masu*)	please	0.657
お/ながい/し/ます (*o/nagai/si/masu*)	お/ねがい/し/ます (*o/negai/shi/masu*)	please	0.590
お/ねがい/し/まふ (*o/negai/shi/mahu*)	お/ねがい/し/ます (*o/negai/shi/masu*)	please	0.678
ちかれ/た (*tikare/ta*)	疲れ/た (*tukare/ta*)	I'm tired	0.742
お/めでとー (*o/medeto*)	お/めでとう (*o/medetou*)	congratulations	0.911
お/めでとぉ (*o/medeto*)	お/めでとう (*o/medetou*)	congratulations	0.796
お/めっとー (*o/metto*)	お/めでとう (*o/medetou*)	congratulations	0.753
お/めてとう (*o/meteto*)	お/めでとう (*o/medetou*)	congratulations	0.665

Table 2: Example of extracted pair of variants-normalization candidates

word segmentations, POS tags, and normal forms for variant words and consist of 7,213 sentences and 995 variant words. We used Unidic (unidic-mecab) [3] as a standard dictionary.

4.2 Baselines and Evaluation Metrics

We compared the three methods listed in Table 3 in our experiments. Conventional means a method that generates no normalization candidates and only uses the word cost and the cost for a pair of adjacent POSs, so we can consider it as a conventional Japanese morphological analysis. Rule-based means the conventional rule-based method proposed by Sasano et al. (2013). The rule-based method considers insertion of long sound symbols and lowercase characters and substitution with long sound symbols and lowercase characters. We basically use the transformation rule of Sasano et al. (2013) and use three features for the model of Rule-based method: the word score, score for a pair of adjacent POSs, and character transformation score. The character transformation score is constant for all transformation rules. Proposed is our method. We used extracted variant-normalization pairs and all features described in 3.2.3.

We evaluated each method on the basis of precision, recall, and the F-value for the overall system accuracy and the recall for normalization. Since Japanese morphological analysis simultaneously estimates the word segmentation and POS tagging, we have to assess whether or not adding the normalization candidates negatively affects a system.

[3] http://osdn.jp/projects/unidic/ (in Japanese)

4.3 Results

4.3.1 Results of Extracted Variant-Normalization Candidates

Table 2 lists examples of the extracted variant-normalization candidates. Our method automatically extracted well-known transformation patterns such as substitution of lowercase characters, insertion of lowercase characters, and insertion of "ー" and "っ" such as "うれしーい (*ureshii*)".

Our method also extracted more variant phonetic transformation patterns such as substitution of "shi" with "pi," "ji," or "hi" and these combinations such as "うれちい (*uretii*)". We also list examples of extracted multi-word variant-normalization pairs. The phrase "お/ねげー/し/ます (*o/nege/shi/masu*)" is a variant pattern of the original phrase "お/ねがい/します (*o/negai/shi/masu*)". Our method extracted these multi-word mappings.

Moreover, our method can extract typing errors such as "お/めでとう (*o/medetou*)" with "お/めてとう (*o/meteto*)" and slang such as "うれしす (*ureshisu*)" with "うれしい (*ureshii).*" Such relatively less frequent patterns were often excluded from normalization targets. Our method also extracts many paraphrases: semantically and phonetically similar pairs that are not variant-normalization

	word seg			word seg and POS tag			normalization
method	prec	rec	F	prec	rec	F	rec
Conventional	0.865	0.949	0.905	0.837	0.919	0.876	-
Rule-based	0.879	0.952	0.914	0.848	0.918	0.881	0.294
Proposed	0.882	0.951	**0.915**	0.851	0.918	**0.883**	**0.340**

Table 3: Test data (Twitter) precision, recall, and F-value results

proposed	conventional	gold	*translation*
(1) おも (思っ) /た *omo* (omott) /*ta*	おもた *omota*	おも (思っ) /た *omo* (omott) /*ta*	I thought
(2) ばよりん (バイオリン) *bayorin* (*baiorin*)	ば/より/ん *ba*/*yo*/*rin*	ばよりん (バイオリン) *bayorin* (*baiorin*)	violin
(3) かんわいい (かわいい) *kanwaii* (*kawaii*)	かん/わ/いい *kan*/*wa* /*ii*	かんわいい (かわいい) *kanwaii* (*kawaii*)	cute
(4) さっむ (寒い) *samu* (*samui*)	さっ/む *sa*/*mu*	さっむ (寒い) *samu* (*samui*)	It is cold
(5) わろえる (笑える) *waroeru* (*waraeru*)	わろ/える *waro*/*eru*	わろえる (笑える) *waroeru* (*waraeru*)	It is funny
(6) おー/こく *oo*/*koku*	おー/こく *oo*/*koku*	おーこく (王国) *ookoku* (*oukoku*)	Kingdom
(7) ついった (ツイート) *tuitta* (*tuitta*)	つ/いっ/た *tu*/*i*/*ta*	ついった (ツイッター) *tuitta* (*tuitta*)	Twitter

"/" indicates the estimated word segmentation. Words in parentheses "()" are estimated normal forms. Underlined words are variant words.

Table 4: Example of morphological analysis and normalization outputs

pairs. This often degrades the results of morphological analysis. We use a discriminative model to prevent such paraphrase pairs appearing in the decoding step.

4.3.2 Morphological Analysis and Normalization Results

Tables 3 and 4 list the results for the Twitter text. The F-value of the proposed method is significantly higher than those of the conventional method and rule-based method. Our method was able to extract broad-coverage variant words, and these candidates also improve the recall of normalization without degrading the overall accuracy of morphological analysis.

Table 4 show examples of the system output. In the table, slashes indicate the positions of the estimated word segmentations, and the correctly analyzed words are written in bold. Examples (1) to (5) are examples improved by using the proposed method. Examples (6) and (7) are examples that were not improved.

Error Analysis There were roughly two types of errors. The first occurred as a result of a lack of variant-normalization candidates, and the second was search errors. Example (6) shows an example of a case in which our method could not generate the correct normalized form because we could not extract the correct normalized form. Because we extract normalization candidates by phrase level, some patterns are difficult to extract as a word unit. To increase recall, we need to extract character-level and morph-level transformation patterns that occur frequently from phrase-level patterns and add them into morphological analysis and normalization. Example (7) shows an example of a case in which a normalized candidate was generated but a search failed. We will need to develop a more complicated model or introduce other features into the current model to reduce the number of search errors.

Besides the above errors, there are some errors in which correct normalization candidates were filtered. In this study, we filtered many candidates to eliminate noise. Some normalization candidates are filtered, and the correct normalization candidates cannot be generated in the word lattice. To increase recall further, we have to filter functions

or calculate similarity scores more precisely. Also, some errors are associated with unknown words. Twitter data contain many unknown words such as names, and our system sometimes treats these names as other nouns. Non-standard and standard words needs to be more precisely discriminated between for higher accuracy.

5 Conclusion and Future Work

We introduced a new idea for extracting variant words from an unsegmented corpus and incorporated it into morphological analysis. The proposed method can effectively analyze noisy words without manual annotation. The limitation of this work is that this method is based on phonetic similarity. Although our method can extract many variant patterns, it cannot extract a pair of words that have quite low phonetic similarity. In addition, our method is based on a heuristic segmentation method for extracting normalization candidates. Though it works well in practice, we want to extend this idea for a more general framework.

In future work, we would like to increase the coverage of variant-normalization pairs. For this, we have to extract the character- and morph-level transformation patterns from the acquired phrase level variant-normalization pairs.

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pages 33–40.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Joint Conference on Empirical Methods in Natural Language Processing*, pages 1–8.

Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text message normalization. *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 71–78.

Bo Han and Timothy Baldwin. 2011. 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*, pages 368–378.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. 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*, pages 421–432.

Hany Hassan and Arul Menezes. 2013. Social text normalization using contextual graph random walks. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1577–1586.

Wenbin Jiang, Haitao Mi, and Qun Liu. 2008. Word lattice reranking for chinese word segmentation and part-of-speech tagging. *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, pages 385–392.

Nobuhiro Kaji and Masaru Kitsuregawa. 2013. Efficient word lattice generation for joint word segmentation and pos tagging in japanese. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 153–161.

Nobuhiro Kaji and Masaru Kitsuregawa. 2014. Accurate word segmentation and pos tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 99–109, Doha, Qatar. Association for Computational Linguistics.

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

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

Chen Li and Yang Liu. 2014. Improving text normalization via unsupervised model and discriminative reranking. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 86–93, Baltimore, Maryland, USA. Association for Computational Linguistics.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written japanese. *Language Resources and Evaluation*, pages 345–371.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. 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*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Deana Pennell and Yang Liu. 2011. A character-level machine translation approach for normalization of sms abbreviations. *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 974–982.

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 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1773–1782, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Akira Sasaki, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui. 2013. Normalization of text in microblogging based on machine learning(in japanese) (in japanese). *In Proc. of The 27th Annual Conference of the Japanese Society for Articial Intelligence*.

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

Yi Yang and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 61–72, Seattle, Washington, USA. Association for Computational Linguistics.