A Simple Approach to Unknown Word Processing in Japanese Morphological Analysis

Ryohei Sasano[1] **Sadao Kurohashi**[2] **Manabu Okumura**[1]

[1] Precision and Intelligence Laboratory, Tokyo Institute of Technology
[2] Graduate School of Informatics, Kyoto University
{sasano,oku}@pi.titech.ac.jp, kuro@i.kyoto-u.ac.jp

Abstract

This paper presents a simple but effective approach to unknown word processing in Japanese morphological analysis, which handles 1) unknown words that are derived from words in a pre-defined lexicon and 2) unknown onomatopoeias. Our approach leverages derivation rules and onomatopoeia patterns, and correctly recognizes certain types of unknown words. Experiments revealed that our approach recognized about 4,500 unknown words in 100,000 Web sentences with only 80 harmful side effects and a 6% loss in speed.

1 Introduction

Morphological analysis is the first step in many natural language applications. Since words are not segmented by explicit delimiters in Japanese, Japanese morphological analysis consists of two subtasks: word segmentation and part-of-speech (POS) tagging. Japanese morphological analysis has successfully adopted lexicon-based approaches for newspaper articles (Kurohashi et al., 1994; Asahara and Matsumoto, 2000; Kudo et al., 2004), in which an input sentence is transformed into a lattice of candidate words using a pre-defined lexicon, and an optimal path in the lattice is then selected. Figure 1 shows an example of a word lattice for morphological analysis and an optimal path. Since the transformation from a sentence into a word lattice basically depends on the pre-defined lexicon, the existence of unknown words, i.e., words that are not included in the pre-defined lexicon, is a major problem in Japanese morphological analysis.

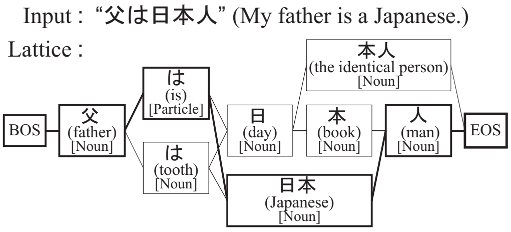

Figure 1: Example of word lattice. The bold lines indicate the optimal path.

There are two major approaches to this problem: one is to augment the lexicon by acquiring unknown words from a corpus in advance (Mori and Nagao, 1996; Murawaki and Kurohashi, 2008) and the other is to introduce better unknown word processing to the morphological analyzer (Nagata, 1999; Uchimoto et al., 2001; Asahara and Matsumoto, 2004; Azuma et al., 2006; Nakagawa and Uchimoto, 2007). Although both approaches have their own advantages and should be exploited cooperatively, this paper focuses only on the latter approach.

Most previous work on this approach has aimed at developing a single general-purpose unknown word model. However, there are several types of unknown words, some of which can be easily dealt with by introducing simple derivation rules and unknown word patterns. In addition, as we will discuss in Section 2.3, the importance of unknown word processing varies across unknown word types. In this paper, we aim to deal with unknown words that are considered important and can be dealt with using simple rules and patterns.

Table 1 lists several types of Japanese unknown words, some of which often appear in Web text. First, we broadly divide the unknown words into two classes: words derived from the words in the lexicon and the others. There are a lot of informal spelling variations in Web text that are derived from the words in the lexicon, such as "ぁなた" (y0u) instead of "あなた" (you) and "冷たーーい" (coooool) instead of "冷たい" (cool). The types of derivation are limited, and thus most of them can be resolved by introducing derivation rules. Unknown words other than those derived from known words are generally difficult to resolve using only simple rules, and the lexicon augmentation approach would be better for them. However, this is not true for onomatopoeias. Although Japanese is rich in onomatopoeias and some of them do not

Unknown words derived from known words		
Type	Unknown word	Original word
*Rendaku** (sequential voicing)	(たまご) ざけ ((tamago-)*zake*, *sake*-nog)	さけ (*sake*, Japanese alcoholic drink)
Substitution with long sound symbols*	ほんとー (troo)	ほんとう (true)
Substitution with lowercases*	ぁなた (y0u)	あなた (you)
Substitution with normal symbols	うれ∪い (h@ppy)	うれしい (happy)
Insertion of long sound symbols*	冷たーーーい (coooool)	冷たい (cool)
Insertion of lowercases*	冷たぁぁぁい (coooool)	冷たい (cool)
Insertion of vowel characters	冷たあああい (coooool)	冷たい (cool)
Unknown words other than those derived from known words		
Type	Unknown word	Corresponding English expression
Onomatopoeia with repetition*	かあかあ	caw-caw
Onomatopoeia w/o repetition*	シュッと	hiss
Rare word / New word	除染 / ツイッター	decontamination / Twitter

Table 1: Various types of Japanese unknown words. The '*' denotes that this type is the target of this research. See Section 2.2 for more details.

appear in the lexicon, most of them follow several patterns such as '*ABAB*,' '*A*っ*B* り,' and '*AB*っと,'[1] and they thus can be resolved by considering typical patterns.

Therefore, in this paper, we introduce derivation rules and onomatopoeia patterns to the unknown word processing in Japanese morphological analysis, and aim to resolve 1) unknown words derived from words in a pre-defined lexicon and 2) unknown onomatopoeias.

2 Background

2.1 Japanese morphological analysis

As mentioned earlier, lexicon-based approaches have been widely adopted for Japanese morphological analysis. In these approaches, we assume that a lexicon, which lists a pair consisting of a word and its corresponding part-of-speech, is available. The process of traditional Japanese morphological analysis is as follows:

1. Build a lattice of words that represents all the candidate sequences of words from an input sentence.
2. Find an optimal path through the lattice.

Figure 1 in Section 1 shows an example of a word lattice for the input sentence "父は日本人" (My father is Japanese), where a total of six candidate paths are encoded and the optimal path is marked with bold lines. The lattice is mainly built with the words in the lexicon. Some heuristics are also used for dealing with unknown words, but in most cases, only a few simple heuristics are used. In fact, the three major Japanese morphological analyzers, JUMAN (Kurohashi and Kawahara, 2005), ChaSen (Matsumoto et al., 2007), and MeCab (Kudo, 2006), use only a few simple heuristics based on the character types, such as hiragana, katakana, and alphabets[2], that regard a character sequence consisting of the same character type as a word candidate.

The optimal path is searched for based on the sum of the costs for the path. There are two types of costs: the cost for a candidate word and the cost for a pair of adjacent parts-of-speech. The cost for a word reflects the probability of the occurrence of the word, and the connectivity cost of a pair of parts-of-speech reflects the probability of an adjacent occurrence of the pair. A greater cost means less probability. The costs are manually assigned in JUMAN, and assigned by adopting supervised machine learning techniques in ChaSen and MeCab, while the algorithm to find the optimal path is the same, which is based on the Viterbi algorithm.

2.2 Types of unknown words

In this section, we detail the target unknown word types of this research.

Rendaku (sequential voicing) is a phenomenon in Japanese morpho-phonology that voices the initial consonant of the non-initial portion of a compound word. In the following example, the initial consonant of the Japanese noun "さけ" (*sake*, alcoholic drink) is voiced into "ざけ" (*zake*):

(1) た ま ご ざ け (eggnog)
ta ma go - za ke.

Since the expression "ざけ" (*zake*) is not included in a standard lexicon, it is regarded as an unknown word even if the original word "さけ" (*sake*) is included in the lexicon. There are a lot

[1] '*A*' and '*B*' denote Japanese characters, respectively.

[2] Four different character types are used in Japanese: *hiragana*, *katakana*, Chinese characters, and Roman alphabet.

of studies on *rendaku* in the field of phonetics and linguistics, and several conditions that prevent *rendaku* are known, such as Lyman's Law (Lyman, 1894), which stated that *rendaku* does not occur when the second element of the compound contains a voiced obstruent. However, few studies dealt with *rendaku* in morphological analysis. Since we have to check the adjacent word to recognize *rendaku*, it is difficult to deal with *rendaku* using only the lexicon augmentation approach.

Some characters are substituted by peculiar characters or symbols such as long sound symbols, lowercase *kana* characters[3], in informal text. First, if there is little difference in pronunciation, Japanese vowel characters 'あ'(a), 'い'(i), 'う'(u), 'え'(e), and 'お'(o) are sometimes substituted by long sound symbols 'ー' or '〜.' For example, a vowel character 'う' in the Japanese adjective "ほんとう" (*hontou*, true) is sometimes substituted by 'ー' and this adjective is written as "ほんとー" (*hontô*, troo). We call this phenomenon **substitution with long sound symbols**. As well as long sound symbol substitution, some *hiragana* characters such as 'あ'(a), 'い'(i), 'う'(u), 'え'(e), 'お'(o), 'わ'(wa), and 'か'(ka) are substituted by their lowercases: 'ぁ,' 'ぃ,' 'ぅ,' 'ぇ,' 'ぉ,' 'ゎ,' and 'ヵ.' We call this phenomenon **substitution with lowercases**.

There are also other types of derivation, that is, some characters are inserted into a word that is included in the lexicon. In the following examples, long sound symbols and lowercase are inserted into the Japanese adjective "冷たい" (cool).

(2) 冷たーーーい (coooool) **(Insertion of long sound symbols)**

(3) 冷たぁぁぁい (coooool) **(Insertion of lowercases)**

In addition to the unknown words derived from words in the lexicon, there are several types of unknown words that contain rare words such as "除染" (decontamination), new words such as "ツイッター" (Twitter), and onomatopoeias such as "かあかあ" (caw-caw). We can easily generate Japanese onomatopoeias that are not included in the lexicon. Most of them follow several patterns, such as '*ABAB*,' '*A*っ*B*り,' and '*AB*っと,' and we classified them into two types, **onomatopoeias with repetition** such as '*ABAB*,' and **onomatopoeias without repetition** such as '*A*っ*B*り.'

2.3 Importance of unknown word processing of each type

The importance of unknown word processing varies across unknown word types.

We give three example sentences (4), (5), and (6), which include the unknown words "もこもこ" (fluffy), "除染" (decontamination), and "ツイッター" (Twitter), respectively. In these examples, (a) denotes the desirable morphological analysis and (b) is the output of our baseline morphological analyzer, JUMAN version 5.1 (Kurohashi and Kawahara, 2005).

(4) Input: ふわふわでもこもこの肌触り。
(A soft and fluffy feeling to the touch.)
(a) ふわふわ / で / もこもこ / の / 肌触り。
soft and fluffy of touch
(b) ふわふわ / でも / こも / この /肌触り。
soft but *straw matting* this touch

(5) Input: 除染が必要。
(Decontamination is required.)
(a) 除染 / が / 必要。
decontamination is required
(b) 除 / 染 / が / 必要。
Unknown Word Unknown Word is required

(6) Input: 昨日、ツイッターを始めた。
(I started Twitter yesterday.)
(a) 昨日、/ ツイッター / を / 始めた。
yesterday Twitter ACC started
(b) 昨日、/ ツイッター / を / 始めた。
yesterday Unknown Word ACC started

In the case of (4), the unknown word "もこもこ" (fluffy) is divided into three parts by JUMAN, and influences the analyses of the adjacent function words, that is, "で" (and) is changed to "でも" (but) and "の" (of) is changed to "この" (this), which will strongly affect the other NLP applications. The wide scope of influence is due to the fact that "もこもこ" consists of *hiragana* characters like most Japanese function words. On the other hand, in the case of (5), although the unknown word "除染" (decontamination) is divided into two parts by JUMAN, there is no influence on the adjacent analyses. Moreover, in case of (6), although there is no lexical entry of "ツイッター" (Twitter), the segmentation is correct thanks to simple character-based heuristics for out-of-vocabulary (OOV) words.

These two unknown words do not contain *hiragana* characters, and thus, we think it is important to resolve unknown words that contain *hiragana*. Since unknown words derived from words in the lexicon and onomatopoeias often contain *hi-*

[3] In this paper, we call the following characters lowercase: 'ぁ,' 'ぃ,' 'ぅ,' 'ぇ,' 'ぉ,' 'ゎ,' and 'ヵ.'

ragana characters, we came to the conclusion that it is more important to resolve them than to resolve rare words and new words that often consist of *katakana* and Chinese characters.

2.4 Related work

Much work has been done on Japanese unknown word processing. Several approaches aimed to acquire unknown words from a corpus in advance (Mori and Nagao, 1996; Murawaki and Kurohashi, 2008) and others aimed to introduce better unknown word model to morphological analyzer (Nagata, 1999; Uchimoto et al., 2001; Asahara and Matsumoto, 2004; Nakagawa and Uchimoto, 2007). However, there are few works that focus on certain types of unknown words.

Kazama et al. (1999)'s work is one of them. Kazama et al. improved the morphological analyzer JUMAN to deal with the informal expressions in online chat conversations. They focused on substitution and insertion, which are also the target of this paper. However, while our approach aims to develop heuristics to flexibly search the lexicon, they expanded the lexicon, and thus their approach cannot deal with an infinite number of derivations, such as "冷たーーい," and "冷ーたーいー" for the original word "冷たい." In addition, Ikeda et al. (2009) conducted experiments using Kazama et al.'s approach on 2,000,000 blogs, and reported that their approach made 37.2% of the sentences affected by their method worse. Therefore, we conjecture that their approach only benefits a text that is very similar to the text in online chat conversations.

Kacmarcik et al. (2000) exploited the normalization rules in advance of morphological analysis, and Ikeda et al. (2009) replaced peculiar expressions with formal expressions after morphological analysis. In this research, we exploit the derivation rules and onomatopoeia patterns in morphological analysis. Owing to such a design, our system can successfully deal with *rendaku*, which has not been dealt with in the previous works.

UniDic dictionary (Den et al., 2008) handles orthographic and phonological variations including *rendaku* and informal ones. However, the number of possible variations is not restricted to a fixed number because we can insert any number of long sound symbols or lowercases into a word, and thus, all the variations cannot be covered by a dictionary. In addition, as mentioned above, since we have to take into account the adjacent word to accurately recognize *rendaku*, the lexical knowledge alone is not sufficient for *rendaku* recognition.

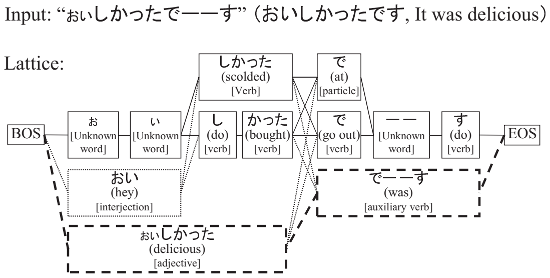
Input: "ぉぃしかったでーーす" (おいしかったです, It was delicious)
Lattice:
BOS
ぉ [Unknown word]
ぃ [Unknown word]
しかった (scolded) [Verb]
し (do) [verb]
かった (bought) [verb]
で (at) [particle]
で (go out) [verb]
ーー [Unknown word]
す (do) [verb]
EOS
おい (hey) [interjection]
ぉぃしかった (delicious) [adjective]
でーーす (was) [auxiliary verb]

Figure 2: Example of a word lattice with new nodes "ぉい," "ぉいしかった," and "でーーす." The broken lines indicate the added nodes and paths, and the bold lines indicate the optimal path.

For languages other than Japanese, there is much work on text normalization that aims to handle informal expressions in social media (Beaufort et al., 2010; Liu et al., 2012; Han et al., 2012). However, their target languages are segmented languages such as English and French, and thus they can focus only on normalization. On the other hand, since Japanese is an unsegmented language, we have to also consider the word segmentation task.

3 Proposed Method

3.1 Overview

We use the rule-based Japanese morphological analyzer JUMAN version 5.1 as our baseline system. Basically we only improve the method for building a word lattice and do not change the process for finding an optimal path from the lattice. That is, our proposed system only adds new nodes to the word lattice built by the baseline system by exploiting the derivation rules and onomatopoeia patterns. If the new nodes and their costs are plausible, the conventional process for finding the optimal path will select the path with added nodes.

For example, if a sentence "ぉいしかったでーーす." is input into the baseline system, it builds the word lattice that is described with solid lines in Figure 2. However, this lattice does not include such expressions as "ぉいしかった" and "でーす" since they are not included in the lexicon. Our proposed system transforms the informal expressions into their standard expressions such as "おいしかった" (delicious) and "です" (was) by exploiting the derivation rules, adds their nodes into the word lattice, and selects the path with these added nodes.

3.2 Resolution of unknown words derived from words in the lexicon

We deal with five types of unknown words that are derived from words in the lexicon: *rendaku*, substitution with long sound symbols, substitution with lowercases, insertion of long sound symbols, and insertion of lowercases. Here, we describe how to add new nodes into the word lattice.

Rendaku The procedure to add unvoiced nodes to deal with *rendaku* differs from the others. Since only the initial consonant of a word is voiced by *rendaku*, there is at most one possible voiced entry for each word in the lexicon. Hence, we add the voiced entries into the trie-based lexicon in advance if the original word does not satisfy any conditions that prevent *rendaku* such as Lyman's Law.

For example, our system creates the entry "ざけ" (zake) from the original word "さけ" (sake), and adds it into the lexicon. When the system retrieves words that start from the fourth character in the example (1) in Section 2.2, "たまござけ," the added entry "ざけ" (zake) is retrieved. Since *rendaku* occurs for the initial consonant of the non-initial portion of a compound word, our system adds the retrieved word only when it is the non-initial portion of a compound word.

Substitution with long sound symbols and lowercases In order to cope with substitution with long sound symbols and lowercases, our system transforms the input text into normalized strings by using simple rules. These rules substitute a long sound symbol with one of the vowel characters: 'あ,' 'い,' 'う,' 'え,' and 'お,' that minimizes the difference in pronunciation. These rules also substitute lowercase characters with the corresponding uppercase characters. For example, if the sentence "ほんとーにぉいしぃ." (It is trooly DElicious.) is input, the nodes generated from the normalized string "ほんとうにおいしい." are added to the word lattice along with the nodes generated from the original string.

Insertion of long sound symbols and lowercases In order to cope with the insertion of long sound symbols and lowercases, our system transforms the input text into a normalized string using simple rules. These rules delete long sound symbols and lowercase characters that are considered to be inserted to prolong the original word pronunciation. For example, if the sentence "冷たぁぁーいでーーーす." (It iiisss coooool.) is input, the nodes generated from the normalized string "冷たいです." are added into the word lattice. We do not consider partly deleted strings such as "冷たぁいでーす." and the combination of substitution and insertion to avoid combinatorial explosion. Therefore, our system cannot deal with unknown words generated by both insertion and substitution, but such words are rare in practice.

Pattern	Example	Transliteration
ABAB	たゆたゆ	tayu-tayu
ABCABC	ぽっかぽっか	pokka-pokka
ABCDABCD	ちょろりちょろり	chorori-chorori

Table 2: Onomatopoeia patterns with repetition and their examples. '*A*,' '*B*,' '*C*,' and '*D*' denote either *hiragana* or *katakana*. We consider only repetitions of two to four characters.

Pattern	Example	Transliteration
H_1つH_2 り	ぽっこり	pokkori
K_1ッK_2 リ	マッタリ	mattari
H_1つH_2Y り	ぺっちゃり	pecchari
K_1ッK_2Y リ	ポッチャリ	pocchari
K_1K_2っと	チラっと	chiratto
K_1K_2ッと	パキッと	pakitto

Table 3: Onomatopoeia patterns without repetition and their examples. '*H*,' denotes the *hiragana*, '*K*' denotes the *katakana*, and '*Y*' denotes the palatalized consonants such as 'ゃ.'

Costs for additional nodes Our system imposes small additional costs to the node generated from the normalized string to give priority to the nodes generated from the original string. We set these costs by using a small development data set.

3.3 Resolution of unknown onomatopoeias

There are many onomatopoeias in Japanese. In particular, there are a lot of unfamiliar onomatopoeias in Web text. Most onomatopoeias follow limited patterns, and we thus can easily produce new onomatopoeias that follow these patterns. Hence, it seems more reasonable to recognize unknown onomatopoeias by exploiting the onomatopoeia patterns than by manually adding lexical entries for them.

Therefore, our system lists onomatopoeia candidates by using onomatopoeia patterns, as shown in Tables 2 and 3, and adds them into the word lattice. Figure 3 shows examples. The number of potential entries of onomatopoeias with repetition is large, but the candidates of onomatopoeias with repetition can be quickly searched for by using a simple string matching strategy. On the other hand, to search the candidates of onomatopoeias without repetition is a bit time consuming com-

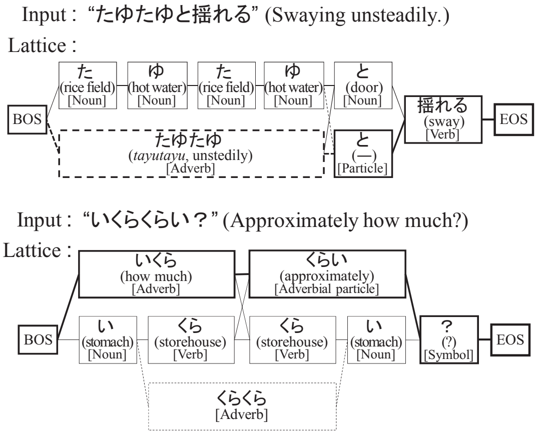

Figure 3: Examples of a word lattice with new nodes of onomatopoeia. The broken lines indicate the added nodes and paths, and the bold lines indicate the optimal path. While the optimal path includes the added node in the upper example, it does not in the lower example.

pared with trie search. However, the number of potential entries of onomatopoeias without repetition is not so large, and thus our system adds all possible entries of onomatopoeias without repetition into the trie-based lexicon in advance.

4 Experiments

4.1 Setting

We used 100,000 Japanese sentences to evaluate our approach. These sentences were obtained from an open search engine infrastructure TSUBAKI (Shinzato et al., 2008), which included at least one *hiragana* character and consisted of more than twenty characters

We first estimated the recall. Since it is too costly to create a set of data with all unknown words annotated, we made a set of data with only our target unknown words annotated. We could apply a set of regular expressions to reduce the unknown word candidates by limiting the type of unknown words. We manually annotated 100 expressions for each type, and estimated the recall.

A high recall, however, does not always imply that the proposed system performs well. It might be possible that our proposed method gives bad effects on non-target words. Therefore, we also compared the whole analysis with and without the rules/patterns from the following seven aspects:[4]

1. The number of positive changes for 100 different outputs: P_{100D}.
2. The number of negative changes for 100 different outputs: N_{100D}.
3. The number of different outputs for 100,000 sentences: D_{100kS}.
4. The estimated number of positive changes for 100,000 sentences: P^*_{100kS}.
5. The estimated number of negative changes for 100,000 sentences: N^*_{100kS}.
6. The relative increase of the nodes: $Node_{inc.}$.
7. The relative loss in speed: SP_{loss}.

Different outputs indicate cases in which the systems with and without rules/patterns output a different result. First, for each type of rule/pattern, we extracted 100 different outputs and manually classified them into three categories: the system with the rules/patterns was better (positive), the system without the rules/patterns was better (negative), and both outputs were undesirable (others). When these outputs differed in word segmentation, we only compared the segmentation but did not take into account the POS tags. On the other side, when these outputs did not differ in word segmentation, we compared the POS tags. Tables 6-10 list several examples. For example, "面白がれる" (can feel amused) in Table 6 should be analyzed as one word, but both systems with and without rules for *rendaku* divided it into several parts, and such a case is labeled as others.

We counted the number of different outputs for 100,000 sentences. We then calculated the estimated numbers of positive/negative changes for the sentences by using the equations:

$$X^*_{100kS} = D_{100kS} \times X_{100D}/100.$$

We also counted the number of created nodes in lattice and calculated the relative increase, which would affect the time for finding the optimal path from the word lattice, and measured the analysis time and calculated the relative loss in speed.

4.2 Results and Discussion

Table 4 lists the recall of our system for each unknown word type with the number of words that are covered by the UniDic dictionary. Note that while our system's recall denotes the ratio of actually recognized words, the coverage of UniDic

[4] There are two major reasons why we did not use the precision, recall and F-measure metrics to evaluate the overall performance. The first reason is that to create a large set of annotated data is too costly. The second reason, which is more essential, is that there is no clear definition of Japanese word segmentation, especially for unknown words. That is, we can accept various word boundaries. We thought it is more straight-forward and efficient to compare the differences between a baseline system and the proposed system.

Unknown word type	Recall of our system	# of words in UniDic
Rendaku (sequential voicing)	83/100	95
Substitution with long sound symbols	99/100	67
Substitution with lowercases	100/100	84
Insertion of long sound symbols	96/100	50
Insertion of lowercases	96/100	73
Onomatopoeia with repetition	89/100	78
Onomatopoeia w/o repetition	94/100	47

Table 4: Recall of our system and the coverage of UniDic.

only denotes the number of words included in the dictionary, which can be interpreted as the upper bound of the system based on UniDic. We can confirm our system achieved high recall for each type of unknown word. Since UniDic covered 95% of unknown words of *rendaku* type, we would be able to improve the *rendaku* recognition by incorporating UniDic and our approach that takes into account the adjacent word. Except for *rendaku*, our system's recall was higher than the coverage of UniDic, which confirms the effectiveness of our method.

Table 5 summarizes the comparison between the analyses with and without the rules/patterns. In short, our method successfully recognized all types of unknown words with few bad effects. By introducing all the derivation rules and onomatopoeia patterns, there are 4,560 improvements for 100,000 sentences with only 80 deteriorations and a 6.2% loss in speed. In particular, the derivation rules of insertion and substitution of long sound symbols and lowercases produced 3,327 improvements for 100,000 sentences at high recall values (see Table 4) with only 27 deteriorations and a 3.8% loss in speed. We confirmed from these results that our approaches are very effective for unknown words in informal text. Since the number of newly added nodes was small, the speed loss is considered to be derived not from the optimal path searching phase but from the lattice building phase.

Table 6 lists some examples of the changed outputs by introducing the derivation rules for *rendaku*. As listed in Table 4 and 5, the *rendaku* processing produced more negative changes and the lower recall value compared with the other types. This indicates that *rendaku* processing is more difficult than resolving informal expressions with long sound symbols or lowercases. Since long sound symbols and lowercases rarely appear in the lexicon, there are few likely candidates other than the correct analysis. On the other hand, voiced characters often appear in the lexicon and formal text, and thus, there are many likely candidates.

Our system	Baseline	Gold standard
Positive		
Input: 洗濯ばさみ (clothespin)		
洗濯/ばさみ	洗濯/ば/さ/み	洗濯/ばさみ
Negative		
Input: 借入れがない方 (the man without)		
借入れ/がない	**借入れ/が/ない**	借入れ/が/ない
Others		
Input: 面白がれる (can feel amused)		
面/白/がれる	面/白/が/れ/る	面白がれる

Table 6: Examples of different outputs by introducing the derivation rule for *rendaku*. The '/' denotes the boundary between words in the corresponding analysis, and the bold font indicates the correct output, that is, the output is the same as the gold standard.

Our approach	Baseline	Gold standard
Positive (insertion)		
Input: 苦〜い経験 (a bitter experiment)		
苦〜い/経験	苦/〜/い/経験	苦〜い/経験
Positive (substitution)		
Input: おめでと〜(congratulations)		
おめでと〜	お/めで/と/〜	おめでと〜
Negative (substitution)		
Input: OK だよ〜ん (It's OK)		
OK/だ/よ〜/ん	OK/だ/よ/〜/ん	OK/だ/よ〜ん
Others (insertion)		
Input: すげー豪華 (very luxury)		
す/げー/豪華	すげ/ー豪華	すげー/豪華

Table 7: Examples of different outputs by introducing derivation rules for long sound symbol substitution and insertion.

Table 7 lists some examples of the changed output by introducing the derivation rules for informal spelling with long sound symbols. We labeled the change of the analysis "OK だよ〜ん" (It's OK) as negative because the baseline system correctly tagged the POS of "だ" unlike our proposed system, but the baseline system could not also correctly resolve the entire phrase. There was no different output that our proposed system could not resolve but the baseline system could fully resolve.

Table 8 lists some examples of the changed outputs by introducing the derivation rules for informal spelling with lowercase. We labeled the change of the analysis "ゆみぃの布団" (Yumi's bedclothes) as negative because the baseline system correctly segmented the postpositional particle "の" unlike our proposed system. Again for this example, the baseline system could not correctly resolve the entire phrase. Along with the informal spelling with long sound symbols, there was no different output that our proposed system could not resolve but the baseline system could fully resolve.

Rules/patterns	P_{100D}	N_{100D}	D_{100kS}	P^*_{100kS}	N^*_{100kS}	$Node_{inc.}$	SP_{loss}
Rendaku (sequential voicing)	37	8	379	140	30	0.553%	2.0%
Substitution with long sound symbols	55	1	920	506	9	0.048%	0.8%
Substitution with lowercases	78	1	1,762	1,374	18	0.039%	0.7%
Insertion of long sound symbols	84	0	1,301	1,093	0	0.038%	1.9%
Insertion of lowercases	88	0	403	354	0	0.019%	0.4%
Onomatopoeia with repetition	74	2	1,162	860	23	0.021%	0.4%
Onomatopoeia w/o repetition	93	0	250	233	0	0.008%	0.0%
Total	-	-	6,177	4,560	80	0.724%	6.2%

Table 5: Comparison between the analyses with and without the rules/patterns.

Our system	Baseline	Gold standard
Positive (insertion)		
Input: 出してくれぃ(please publish)		
出して/くれぃ	出して/くれ/ぃ	出して/くれぃ
Positive (substitution)		
Input: おにぃちゃん (big brother)		
お/にぃちゃん	お/に/ぃ/ちゃん	お/にぃちゃん
Negative (substitution)		
Input: ゆみぃの布団 (Yumi's bedclothes)		
ゆみ/ぃの/布団	ゆみ/ぃ/の/布団	ゆみぃ/の/布団
Others (insertion)		
Input: さみすぃ(lonely)		
さ/みすぃ	さ/みす/ぃ	さみすぃ

Table 8: Examples of different outputs by introducing derivation rules for lowercase substitution and insertion.

Our system	Baseline	Gold standard
Positive		
Input: たゆたゆと (wavy)		
たゆたゆ/と	た/ゆ/た/ゆ/と	たゆたゆ/と
Negative		
Input: あらあら (wow wow)		
あらあら	**あら/あら**	あら/あら

Table 9: Examples of different outputs by introducing onomatopoeia patterns with repetition.

Our system	Baseline	Gold standard
Positive		
Input: ぺっちゃり (flat)		
ぺっちゃり	ぺ/っちゃ/り	ぺっちゃり
Input: チラっと (at a glance)		
チラっと	チラ/っと	チラっと

Table 10: Examples of different outputs by introducing onomatopoeia patterns without repetition.

Table 9 lists some examples of the changed outputs by introducing onomatopoeia patterns with repetition. Our system recognized unknown onomatopoeias with repetition at a recall of 89%, which is not very high. However, since there were several repetition expressions other than onomatopoeias, such as "あら/あら" (wow wow) as shown in Table 9, we cannot lessen the cost for onomatopoeias with repetition.

Table 10 lists some examples of the changed outputs by introducing onomatopoeia patterns without repetition. Our system recognized the unknown onomatopoeias without repetition at a recall of 94% and did not output anything worse than the baseline output with no loss in speed.

Type	# of types	# of tokens
Covered by Murawaki's Lexicon	13	51
Covered by Wikipedia	68	407
Covered by our method	15	105
Others	22	82
Total	118	645

Table 11: Classification results of unknown words that occur more than two times in KNB corpus.

In order to approximate the practical coverage of our method, we classified unknown words that occur more than two times in the Kyoto University and NTT Blog (KNB) corpus[5] into four types: words that are covered by the lexicon created by Murawaki and Kurohashi (2008) (Murawaki's Lexicon), words that are not covered by Murawaki's Lexicon but have entries in Wikipedia, words that are covered only by our method, and the others. Table 11 shows the results. There are total 645 tokens of unknown words that occur more that two times in KNB corpus, 105 of which are newly covered by our method. Since the number of tokens that are covered by neither Murawaki's Lexicon nor Wikipedia is only 187, we can say that the coverage of our method is not trivial.

5 Conclusion

We presented a simple approach to unknown word processing in Japanese morphological analysis. Our approach introduced derivation rules and onomatopoeia patterns, and correctly recognized certain types of unknown words. Our experimental results on Web text revealed that our approach could recognize about 4,500 unknown words for 100,000 Web sentences with only 80 harmful side effects and a 6% loss in speed. We plan to apply our approach to machine learning-based morphological analyzers, such as MeCab, with UniDic dictionary, which handles orthographic and phonological variations, in future work.

[5] The KNB corpus consists 4,186 sentences from Japanese blogs, and is available at http://nlp.kuee.kyoto-u.ac.jp/kuntt/.

References

Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proc. of COLING'00*, pages 21–27.

Masayuki Asahara and Yuji Matsumoto. 2004. Japanese unknown word identification by character-based chunking. In *Proc. of COLING'04*, pages 459–465.

Ai Azuma, Masayuki Asahara, and Yuji Matsumoto. 2006. Japanese unknown word processing using conditional random fields (in Japanese). In *Proc. of IPSJ SIG Notes NL-173-11*, pages 67–74.

Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proc. of ACL'10*, pages 770–779.

Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. 2008. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In *Proc. of LREC'08*, pages 1019–1024.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proc.of EMNLP-CoNLL'12*, pages 421–432.

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

Gary Kacmarcik, Chris Brockett, and Hisami Suzuki. 2000. Robust segmentation of japanese text into a lattice for parsing. In *Proc. of COLING'00*, pages 390–396.

Jun'ichi Kazama, Yutaka Mitsuishi, Makino Takaki, Kentaro Torisawa, Koich Matsuda, and Jun'ichi Tsujii. 1999. Morphological analysis for japanese web chat (in Japanese). In *Proc. of 5th Annual Meetings of the Japanese Association for Natural Language Processing*, pages 509–512.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proc. of EMNLP'04*, pages 230–237.

Taku Kudo, 2006. *MeCab: Yet Another Part-of-Speech and Morphological Analyzer*. http://mecab.sourceforge.jp/.

Sadao Kurohashi and Daisuke Kawahara. 2005. Japanese morphological analysis system JUMAN version 5.1 manual.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, , and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proc. of The International Workshop on Sharable Natural Language Resources*, pages 22–38.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proc. of ACL'12*, pages 1035–1044.

Benjamin Smith Lyman. 1894. *The change from surd to sonant in Japanese compounds*. Philadelphia : Oriental Club of Philadelphia.

Yuji Matsumoto, Kazuma Takaoka, and Masayuki Asahara. 2007. Chasen: Morphological analyzer version 2.4.0 user's manual.

Shinsuke Mori and Makoto Nagao. 1996. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proc. of COLING'96*, pages 1119–1122.

Yugo Murawaki and Sadao Kurohashi. 2008. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Proc. of EMNLP'08*, pages 429–437.

Masaaki Nagata. 1999. A part of speech estimation method for japanese unknown words using a statistical model of morphology and context. In *Proc. of ACL'99*, pages 277–284.

Tetsuji Nakagawa and Kiyotaka Uchimoto. 2007. A hybrid approach to word segmentation and pos tagging. In *Proc. of ACL'07*, pages 217–220.

Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. Tsubaki: An open search engine infrastructure for developing new information access methodology. In *Proc. of IJCNLP'08*, pages 189–196.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. The unknown word problem: a morphological analysis of japanese using maximum entropy aided by a dictionary. In *Proc. of EMNLP'01*, pages 91–99.