

Expanding Vocabulary for Recognizing User's Abbreviations of Proper Nouns without Increasing ASR Error Rates in Spoken Dialogue Systems

Masaki Katsumaru, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno

Graduate School of Informatics, Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan.

{katumaru, komatani, ogata, okuno}@kuis.kyoto-u.ac.jp

Abstract

Users often abbreviate long words when using spoken dialogue systems, which results in automatic speech recognition (ASR) errors. We define *abbreviated words* as sub-words of the original word, and add them into an ASR dictionary. The first problem is that proper nouns cannot be correctly segmented by general morphological analyzers, although long and compounded words need to be segmented in agglutinative languages such as Japanese. The second is that, as vocabulary increases, adding many abbreviated words degrades the ASR accuracy. We develop two methods, (1) to segment words by using conjunction probabilities between characters, and (2) to manipulate occurrence probabilities of generated abbreviated words on the basis of the phonological similarities between abbreviated and original words. By our method, the ASR accuracy is improved by 24.2 points for utterances containing abbreviated words, and degraded by only a 0.1 point for those containing original words.

Index Terms: spoken dialogue systems, abbreviated words, proper nouns, vocabulary expansion

1. Introduction

Users often omit a part of long words and utter abbreviated words [1]. They are apt to do this, because users unfamiliar with a particular spoken dialogue system do not know much about how to use it and what content words are included in its vocabulary. In conventional system developments, system developers manually add unknown words to the ASR dictionary by collecting and examining misrecognized words uttered by users. The manual maintenance requires a great deal of time and effort. Furthermore, a system cannot recognize these words until the manual maintenance has taken place. They continue to be misrecognized until the system developers find and add them into the system dictionary.

Our purpose is to automatically add abbreviated words users may utter at the initial time when an original dictionary in any domains has been provided. An *original dictionary* is defined as the initial ASR dictionary that a system has originally, *original words* as content words in an original dictionary, and *abbreviated words* as those that are sub-words of an original word and that also indicate the same entity as the original word. We generate abbreviated words by omitting arbitrary sub-words of an original word. Automatic addition of vocabulary at the initial stage of system developments alleviates manual maintenance time and efforts. Furthermore, the system can recognize abbreviated words at an earlier stage, thus increasing usability of the system.

There are two issues when abbreviated words are added into an ASR dictionary.

1. Segmenting proper nouns in order to generate abbreviated words

Proper nouns cannot be correctly segmented by general morphological analyzers because they are domain-dependent words, such as regional names. To decide which sub-words to omit, segment action of proper nouns is needed in agglutinative languages such as Japanese, while words in an isolated language such as English do not pose this problem.

2. Reducing ASR errors caused by adding abbreviated words into an ASR dictionary

The ASR accuracy often degrades by adding generated abbreviated words because the size of vocabulary increases. Jan et al. merely added generated abbreviated words and did not take the degradation into account [2]. Abbreviated words whose phonemes are close to those of other original words are likely to be confused in the ASR.

For the former, we segmented proper nouns by using conjunction probabilities between characters in addition to the results of a morphological analyzer. For the latter, we manipulate occurrence probabilities of generated abbreviated words on the basis of the phonological similarities between the abbreviated and original words. Thus, we can add abbreviated words into an ASR dictionary without increasing ASR error rates.

2. Case study of deployed system

We preliminarily investigated gaps between users' utterances and the vocabulary of a system by analyzing words added by developers during the 5-year service of the Kyoto City Bus Information System [3]. Users input their boarding stop as well as the destination or the bus route number by telephone, and the system tells them how long it will be before the bus arrives. There were 15,290 calls to the system during the 58 months between May 2002 and February 2007, and the system developers added users' words that the system could not recognize¹.

The developers added 309 words to the system's vocabulary. Of these 91.6% were alias names for the same entity, while 8.4% were new entities of bus stops and landmarks. There were far fewer new entities added than alias names for the same entity. This means that the developers had carefully prepared the vocabulary for bus stops and landmarks at the initial stage of

¹The developers did not add all words users uttered during this period. Short words were not added because they may cause insertion errors. This was because the system's dialogue management is executed in a mixed-initiated manner, and its language constraint is not so strong.

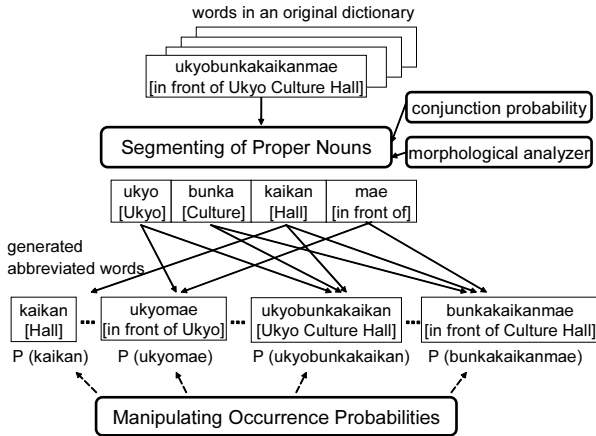


Figure 1: Flow of adding abbreviated words

system development. The reason why added words consist almost exclusively of alias names is that at the initial stage of system development, the system’s developers were unable to predict the wide range of other expressions uttered by real users. The most added alias names are abbreviated words, which were 78.3% of all added words. This means that real users often utter abbreviated words actually. Of the 1,494 utterances collected from novices using the system, 150 utterances contained abbreviated words.

Getting abbreviated words automatically from an outside information source such as corpora or the Web [4, 5] is difficult in spoken dialogue systems because the abbreviated words are domain-dependent proper nouns and their frequency of appearance in the Web is generally low. The frequency of appearance in the Web of the actual added abbreviated words was actually very low. For example, the frequency of “shinrinkodanjutaku” (Name of Place) was 2, which is far smaller than word fragments with no meaning.

3. Generating and manipulating occurrence probabilities of abbreviated words

The flow of our method of adding abbreviated words is shown in Figure 1. First, we segment original words to decide sub-words to omit. For domain-dependent proper nouns, we define a *conjunction probability* between each character as the measure of segmenting compound words. In section 3.1, we segment proper nouns by using conjunction probabilities and a morphological analyzer. Then we generate abbreviated words by omitting some sub-words of segmented words. In section 3.2, we suppress ASR errors caused by adding generated abbreviated words whose phonemes are close to those of words in the original ASR dictionary. We define a phonological similarity between abbreviated and original words, and manipulate occurrence probabilities on the basis of the similarities. Thus, we reduce ASR errors caused by adding generated abbreviated words. These methods are independent of a system’s domain.

Segment result of MeCab:

kokusai	kaikan	ekimae
[International]	[Hall]	[in front of Station]

Segment result of conjunction probabilities

kokusaikaikan	eki	mae
[International Hall]	[Station]	[in front of]

Segment result of MeCab and conjunction probabilities:

kokusai	kaikan	eki	mae
[International]	[Hall]	[Station]	[in front of]

Figure 2: Segmenting of “kokusaikaikanekimae”

3.1. Segmenting words in the ASR dictionary and generating abbreviated words

First, we segment a compound word in an ASR dictionary into a sub-word array “ $s_1s_2 \dots s_n$ ”. We segment at a part where either a morphological analyzer or conjunction probabilities segments. The morphological analyzer we use is MeCab [6]. Segmenting domain-dependent proper nouns by using conjunction probabilities between characters is described as follows. When a word in an ASR dictionary is expressed by a character string “ $c_1c_2 \dots c_{i-1}c_ic_{i+1} \dots c_n$ ”, a conjunction probability between c_{i-1} and c_i is formulated based on the characteristic N-gram probabilities in an ASR dictionary as follows.

$$\min\{P(c_i|c_{i-1}c_{i-2} \dots c_1), P(c_{i-1}|c_ic_{i+1} \dots c_n)\} \quad (1)$$

This means that a conjunction probability is defined as smaller one of N-gram probabilities forward to c_i and backward to c_{i-1} . We segment a word between c_{i-1} and c_i if the conjunction probability between them is lower than threshold θ . For example, a proper noun “kokusaikaikanekimae”, which stands for “in front of International Hall Station”, is segmented as shown in Figure 2. Using conjunction probabilities can segment “ekimae” between “eki” and “mae”, while using MeCab cannot do so. This segmentation is essential to generate various abbreviated words such as “kokusaikaikaneki” which was in fact uttered by a user.

Next, we omit an arbitrary number of sub-words and generate $(2^n - 1)$ abbreviated words from a sub-word array “ $s_1s_2 \dots s_n$ ”. We give pronunciations of generated abbreviated words by pronunciations of sub-words, which are detected by matching between pronunciations which MeCab gives and an original pronunciation.

3.2. Reducing ASR errors caused as a result of adding generated abbreviated words

3.2.1. Definition of phonological similarity

We define a phonological similarity as a measure of confusion in ASR that is caused by generated abbreviated words. These words will cause ASR errors for utterances containing original words in the case when phonemes of added words are close to those of original words or those of only part of original words. We define a phonological similarity between a generated abbreviated word, w , and vocabulary, D_{org} , of an original dictionary as follows.

$$dist(w, D_{org}) = \min(e.d.(w, part(D'_{org})) \quad (2)$$

Table 1: Abbreviated words whose phonemes are close to those of parts of original words in Kyoto City Bus Information System

abbreviated word (before abbr.)	original word	<i>e.d.</i>
noda (nodacho) [Name of Town]	hanazonodaigaku [Hanazono University]	0
kamitoba (kamitobatonomori) [Name of Area]	kaminobashi [Kamino Bridge]	1
shakadani (shakadaniguti) [Name of Area]	haradani [Name of Area]	2

We denote D'_{org} as vocabulary that is made of D_{org} by removing words from which we generate w . The partial sequences of all words of D'_{org} is given by $part(D'_{org})$. The edit distance between x 's and y 's phoneme strings is $e.d.(x, y)$, and we calculate it by DP matching [7]. When we define S_1 as a phoneme set of vowels, a moraic obstruent and a moraic nasal, and S_2 as that of consonants, we set costs of the edit distance 2 when an element of S_1 is inserted, deleted, or substituted, and of 1 when an element of S_2 is inserted, deleted, or substituted with one of S_2 .

We generate abbreviated words from the vocabulary of the Kyoto City Bus Information System. Various generated abbreviated words whose phonemes are close to those of other original words are shown in Table 1. The phonological similarity between a generated abbreviated word, “noda”, and vocabulary of the original dictionary is 0 because “noda” is equal to part of “hanazonodaigaku”.

3.2.2. Manipulating occurrence probabilities on the basis of phonological similarity

In order to avoid degrading the ASR accuracy for utterances containing original words, we manipulate occurrence probabilities of generated abbreviated words on the basis of the phonological similarities. For example, the system will recognize an original word “kaminobashi” correctly by reducing an occurrence probability of a generated abbreviated word “kamitoba”. We use a statistical language model to manipulate the occurrence probability of each word.

We define $P_{org}(w)$ as an occurrence probability of a word w . The generated abbreviated words that meet condition:

$$dist(w, D_{org}) \leq d : threshold \quad (3)$$

are arranged as a new occurrence probability $P_{new}(w)$ as follows.

$$P_{new}(w) = P_{org}(w) * \alpha^{dist(w, D_{org}) - d - 1} \quad (4)$$

We set α as 10. The lower this phonological similarity, the lower the occurrence probability arranged. We normalize the probabilities of original and generated abbreviated words after calculating $P_{new}(w)$ of all generated abbreviated words.

4. Experimental evaluation

We have conducted experiments to evaluate our method. We generated abbreviated words from a system’s ASR dictionary and added them into the dictionary. Measures for evaluations are the recall rate and the ASR accuracy for collected utterances. Furthermore, to demonstrate that our method is independent of a particular domain, we also generated abbreviated words in another domain.

Table 2: Recall rate [%] by each method of segmenting

Method of segmentation	Recall rate
conjunction probability only	73
MeCab (morphological analyzer) only	86
MeCab + conjunction probability (our method)	94

4.1. Target data for evaluation

We evaluated our method by using real users’ utterances collected on the Kyoto City Bus Information System. As target was the users who lack knowledge of the system’s vocabulary, we collected utterances of novices who have used the system only once by analyzing their telephone numbers. We collected 1,494 utterances by 183 users after removing utterances that are not related to the task. Of the 1,494 utterances, 150 ones contain 70 kinds of abbreviated words of original words, 1,142 ones contain only original words that the system can recognize in the original dictionary. The others 202 ones consist of words which are neither abbreviated nor original words, such as “Does this system tell me where to change buses?”.

4.2. Recall rate of generated abbreviated words

We generated abbreviated words from 1,481 content words (bus stops, landmarks, and bus route numbers) of 1,668 original words of the Kyoto Bus Information System. The threshold of segmentation by conjunction probabilities were set at 0.12 after preliminary experiments. We segmented words by using both conjunction probabilities and MeCab, omitted sub-words, and generated 11,936 abbreviated words. To prove effectiveness of our segmentation, we also generated 2,619 abbreviated words by segmentation of only conjunction probability, 8,941 abbreviated words by segmentation of only MeCab. We evaluated three methods of segmentation:

- conjunction probability only
- MeCab (morphological analyzer) only
- both conjunction probability and MeCab (our method)

The recall rates of abbreviated words generated by each method are shown in Table 2. For 70 kinds of abbreviated word uttered by users, our method generated 66 kinds (94%) while 51 kinds (73%) were generated by only conjunction probability and 60 kinds (86%) by using only MeCab. The recall rate of our method was 8 points higher than that produced by only MeCab. Segmenting proper nouns by using conjunction probability led to this improvement.

4.3. Evaluation of ASR Accuracy

We made the statistical language model in order to manipulate occurrence probabilities for each word because Kyoto City Bus Information System’s ASR is a grammar-based one. First, content words were assigned to the class of bus stops and landmarks or that of bus route numbers. Next, we made the class N-gram model from all kinds of sentences that the grammar-based language model generated. We used CMUToolkit [8] to construct the statistical language model. We added the abbreviated words generated to the class of bus stops and landmarks in addition to original bus stops or landmarks. The acoustic model is a triphone model of 2,000 states and 16 mixture components for telephone speech. The ASR engine is Julius [9]. We set a value

Table 3: ASR accuracy [%] for content words of utterances in each condition

Conds.	utterances with abbreviated words	utterances with original words	utterances of all
#1	1.1	74.9	52.5
#2	24.7	59.8	40.2
#3	25.3	74.8	56.5
#4	49.5	74.2	58.2

of $d=5$ by trial and error. The experimental conditions were as follows:

Cond. #1: original dictionary (baseline)

Use an ASR dictionary of the system before adding abbreviated words (vocabulary size: 1,668)

Cond. #2: original dictionary + generated abbreviated words

Add generated abbreviated words into the original dictionary (vocabulary size: 13,604)

Cond. #3: original dictionary + generated abbreviated words + manipulate probabilities (our method)

Add generated abbreviated words into the original dictionary and manipulate occurrence probabilities (vocabulary size: 13,604)

Cond. #4: original dictionary + 70 kinds of abbreviated words (upper limit)

Add 70 kinds of abbreviated words actually uttered by users into the original dictionary. The ASR accuracy in this condition is the upper limit of our method. (vocabulary size: 1,738)

Table 3 shows ASR accuracy of content words for “utterances with abbreviated words”, which are 150 utterances containing abbreviated words; “utterances with original words”, which are 1,142 utterances containing only original words; and “utterances of all”, which are 1,494 utterance of all.

The ASR accuracy for the utterances with abbreviated words from the original dictionary was 1.1%. By merely adding abbreviated words generated by our method, the ASR accuracy improved by 23.6 points. However, the ASR accuracy for utterances with original words degraded by 15.1 points. Thus, the ASR accuracy for all utterances degraded by 12.3 points compared to the original dictionary. The results shows that the ASR accuracy degrades by merely adding vocabulary. In our method, the ASR accuracy for utterances with original words improved by 15.0 points compared to that gained merely adding, and it degraded by only a 0.1 point compared to that obtained by using the original dictionary. These results also demonstrate the effectiveness of our method of manipulating occurrence probabilities for reducing ASR errors caused by adding vocabulary. The ASR accuracy is still low totally. Even in Cond. #4, which is the upper limit of our method for all utterances, the ASR accuracy for all utterances is 58.2%. A reason for this low level of accuracy is a mismatch between the acoustic model and the users’ circumstances. Acoustic scores of some words the users utter are lower than those of other words. We have addressed how to improve language model, but it is expected that improving the acoustic model will lead to a higher level of ASR accuracy.

4.4. Generating abbreviated words in another domain

We also generated abbreviated words in the restaurant domain to demonstrate that our method is independent of a particular

domain. In this domain, we also correctly segmented domain-dependent proper nouns by using conjunction probabilities, and we generated several appropriate abbreviated words, although a morphological analyzer cannot segment some of them. For example, our method could segment “bisutorokyatorudoru” between “bisutoro (bistro)” and “kyatorudoru (a name of restaurants)” by detecting high frequency of “bisutoro” in the dictionary, although MeCab could not segment it. This segmentation enabled us to generate an abbreviated word “kyatorudoru”, which is often used.

5. Conclusion

We generated abbreviated words and added them into an ASR dictionary to recognize abbreviated words uttered by users. To increase the recall rate of generated abbreviated words, we segment proper nouns by introducing conjunction probabilities between characters in the system’s dictionary. To add abbreviated words without increasing ASR error rates, we reduce their occurrence probabilities on the basis of the phonological similarity between abbreviated and original words.

Experimental evaluations by real users’ utterances demonstrated our method’s effectiveness. The recall rate by our method was higher than that gained by using only a morphological analyzer. The ASR accuracy gained with our method for utterances with abbreviated words was 24.2 points higher than that gained with original dictionary. For utterances containing original words, the ASR accuracy of our method degraded by only a 0.1 point from an original dictionary, while a method in which generated abbreviated words were merely added into the ASR dictionary degraded the ASR accuracy by 15.1 points from when an original dictionary was used. These results show that our method of vocabulary expansion can recognize user’s abbreviated words without increasing ASR error rates.

There are many kinds of alias names that indicate original words, such as aliases that are changed the morpheme order in a word, as well as abbreviated words. We can make a robust ASR by adding these expressions into an ASR dictionary.

6. References

- [1] G. Zweig, P. Nguyen, Y. Ju, Y. Wang, D. Yu, and A. Acero, “The Voice-Rate Dialog System for Consumer Ratings,” in *Proc. Interspeech*, 2007, pp. 2713–2716.
- [2] E. E. Jan, B. Maison, L. Mangu, and G. Zweig, “Automatic Construction of Unique Signatures and Confusable Sets for Natural Language Directory Assistance Applications,” in *Proc. Eurospeech*, 2003, pp. 1249–1252.
- [3] K. Komatani, S. Ueno, T. Kawahara, and H. G. Okuno, “User Modeling in Spoken Dialogue Systems for Flexible Guidance Generation,” in *Proc. Eurospeech*, 2003, pp. 745–748.
- [4] Y. Park and R. J. Byrd, “Hybrid text mining for finding terms and their abbreviations,” in *Proc. EMNLP*, 2001, pp. 126–133.
- [5] N. Sundaresan and J. Yi, “Mining the Web for relations,” *Computer networks*, vol. 33, pp. 699–711.
- [6] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying conditional random fields to Japanese morphological analysis,” in *Proc. EMNLP*, 2004, pp. 230–237, <http://mecab.sourceforge.net/>.
- [7] G. Navarro, “A Guided Tour to Approximate String Matching,” *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.
- [8] P. R. Clarkson and R. Rosenfeld, “Statistical Language Modeling Using the CMU-Cambridge Toolkit,” in *Proc. ESCA Eurospeech*, 1997, pp. 2707–2710, <http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html>.
- [9] T. Kawahara, A. Lee, K. Takeda, K. Itou, and K. Shikano, “Recent progress of open-source LVCSR Engine Julius and Japanese model repository,” in *Proc. ICSLP*, 2004, pp. 3069–3072.