3B1-OS22b-1

SPOKEN INTERFACE FOR CORRECTING PHONEME RECOGNITION ERRORS IN LEARNING OF UNKNOWN WORDS

Xiang Zuo[*1] Taisuke Sumii[*1] Naoto Iwahashi[*2] Mikio Nakano[*3] Kotaro Funakoshi[*3]
Natsuki Oka[*1]

[*1] Kyoto Institute of Technology, Japan

[*2] National Institute of Information and Communications Technology, Japan

[*3] Honda Research Institute Japan Co., Ltd, Japan

This paper describes a novel method that enables users to teach systems the phoneme sequences of new words through speech interaction. Using the method, users can correct mis-recognized phoneme sequences incrementally by making *corrective utterances*. Each corrective utterance may include the whole or a segment of the word. During the interaction, if the correction using the utterance results in a better phoneme sequence than the previous one, a user can stop the interaction or make a corrective utterance again. Otherwise the user can reject the utterance. The originalities of this method are 1) interactive correction by speech, 2) the use of spoken word segments for locating mis-recognized phonemes and, 3) the use of generalized posterior probability (GPP) as a measure of correcting mis-recognized phonemes. The experimental results show that the proposed method achieved 96.8% in phoneme accuracy and 79.1% in word accuracy, with less than seven corrective utterances.

1.. INTRODUCTION

One of the main difficulties with conversational robots in the physical world is the learning of out-of-vocabulary (OOV) words. A conversational robot to be used in a home environment may encounter an object with a name that does not exist in its vocabulary. To recognize and synthesize such names, the robot should have the capability of learning the correct phoneme sequences of new words.

Previously, several word learning methods, which extract OOV words from spontaneous utterances, have been proposed, such as [1]. However, these methods did not focus on improving the accuracy of the recognized phoneme sequences of OOV words. It is very difficult because state-of-the-art speech recognition systems do not achieve adequate performance in phoneme recognition [2].

To improve phoneme accuracy, there have been a number of studies in the context of automatic phonetic transcription. Jain et al. [3] proposed a method to improve phoneme recognition accuracy by creating speaker-specific phonetic templates. Itou et al. [4] developed a method to improve the phoneme accuracy of OOV words by calculating the average likelihood of the recognized phoneme sequences of the speech samples obtained from multiple persons. Bael et al. [5] compared the applicability of ten procedures for the automatic phonetic transcription of large-scale speech corpus. Taguchi et al. [6] proposed a method of learning the correct phoneme sequences of OOV words based on statistical model selection by integrating information obtained not only from spoken utterances but also from their meanings. In these previous methods, the processes were carried out in off-line learning without any interactions with humans.

On the other hand, on-line phonetic transcription from a small number of utterances is a difficult task, so some of the previous methods ask the user to spell out the new word [7, 8]. However, spelling out is not effective in languages such as Japanese or Chinese.

In this study, we propose an interactive method for learning the phoneme sequences of new words, that allows users to make corrective utterances to correct phoneme recognition errors. The following dialog scenario between a user (U) and a system (S) shows an example of the target task of this study.

U: My name is Taisuke Sumii.
S: **Taizuke Sumie**?
U: No. Taisuke Sumii.
S: **Taizuke** Sumii?
U: No. Listen, Taisuke.
S: Taisuke Sumii?
U: That's right.

Here, the user first tries to teach the system her/his name "Taisuke Sumii" by an utterance. The system mis-recognizes certain phonemes for the name. The user corrects them by making utterances. In this task, notice that rather than repeating the full name, the user might also make a corrective utterance, just repeating a part of the word according to the error in the recognized phoneme sequence. This kind of partial correction may often be found in dialogs between humans.

This paper proposes the method that aims at realization of such a dialog for learning new words. The originalities of the proposed method are summarized as follows:

Spoken interactive correction: The correction process is run in an interactive way, rather than in a batch way, which makes the correction more efficient.

Word segment error locating: Apart from the whole word, the user can also use word segments in a corrective utterance for locating mis-recognized phonemes to prevent mis-correction of the correct part of the phoneme sequence.

GPP-based phoneme correction: A generalized posterior probability (GPP) is used as a measure for correcting mis-recognized phonemes.

This paper is organized as follows. Section 2. describes the interactive phoneme correction algorithm. The experimental methodology and results are presented in Section 3.. Finally, Section 4. concludes the paper.

Contact: Xiang Zuo, d8821502@edu.kit.ac.jp

```
1. Set $i \leftarrow 0$, $M \leftarrow$ maximum number of correction.
2. Get a phoneme sequence $x_0$ of OOV word from an initial
   utterance $u_0$.
3. Set $y_0 \leftarrow x_0$, and reponses user by $y_0$.
    4. Depending on user's reponse, switches between
       [stop mode]: go to step 11,
       [progress mode]: go to step 5.
        5. $i \leftarrow i + 1$.
        6. If $i > M$ then go to step 13, otherwise go to step 7.
            7. Get a phoneme sequence $x_i$ of OOV word from a
               corrective utterance $u_i$.
            8. Use $x_i$ to correct phoneme errors in $y_{i-1}$ by word
               segment error locating and GPP-based phoneme
               correction.
            9. Set $y_i \leftarrow$ correction result, and output $y_i$ to user.
                10. According to $y_i$, user makes an utterance to
                    switch among
                    [stop mode]: go to step 11,
                    [progress mode]: go to step 5,
                    [return mode]: set $y_i \leftarrow y_{i-1}$, and go to step 5.
11. Stop the correction process, and output $y_i$.
```

Fig. 1. The interactive phoneme correction algorithm.

2.. INTERACTIVE PHONEME ERROR CORRECTION ALGORITHM

The proposed interactive phoneme error correction algorithm is shown in Fig. 1. First, a user makes an initial utterance u_0 as "this is <*oov*>" to teach the system a new word. Here, <*oov*> denotes the new word (OOV word) in u_0. The system gets a phoneme sequence x_0 of the OOV word from u_0 by a HMM-based phoneme recognizer using a pre-defined grammar. The system then assigns x_0 to an output phoneme sequence y_0, and responds to the user by outputting y_0. According to y_0, the user makes an utterance to switch between a stop mode and a progress mode. In the progress mode, the user makes a corrective utterance u_1. Then iterative process begins. In the i-th iteration, the phoneme sequence x_i of the OOV word in u_i is extracted as the same way of x_0. The system then uses x_i to correct phoneme errors in phoneme sequence y_{i-1}. The correction process is performed by word segment error locating and GPP-based phoneme correction. The correction result is assigned to y_i, and the system responds the user by outputting y_i. According to y_i, the user makes an utterance to switch among a stop mode, a progress mode and a return mode. The details of them are given in Section 2.1. This iterative process is repeated until a correct phoneme sequence is output by the system, or the number of corrective utterances becomes the maximum number M.

In this paper, we call u_0 *initial utterance*, $u_i(i = 1, \ldots, M)$ *corrective utterance*, $x_i(i = 1, \ldots, M)$ *corrective phoneme sequence*, and $y_i(i = 0, \ldots, M)$ *output phoneme sequence*. The remaining parts of this section give details about the three modes in above-described interactive correction process, the word segment error locating, and the GPP-based phoneme correction.

2.1. Spoken interactive correction

The spoken interactive correction is consisted of a stop mode, a progress mode and a return mode, each of which is defined as follows:

Stop mode: If a user considers that y_i is a correct phoneme sequence, she/he can stop the correction process by an utterance as "that's right."

y

		𝕊	g	a	sh	i	r	a	o	n	o	r	i	𝔼
	𝕊	0.0	1.5	3.0	4.5	6.0	7.5	9.0	10.5	12.0	13.5	15.0	16.5	18.0
	sh	1.5	6.6	12.4	3.3	9.9	16.9	18.3	15.2	15.9	15.3	24.6	24.1	19.5
	i	3.0	5.4	8.8	7.4	3.4	8.2	10.9	16.0	18.9	21.3	22.3	20.4	21.0
	b	4.5	9.1	5.2	14.1	6.0	6.3	15.6	15.5	11.1	19.2	23.2	28.8	22.7
	a	6.0	9.4	6.4	8.9	11.4	10.3	6.7	10.0	17.5	22.6	24.6	28.3	23.4
x	k	7.5	13.4	6.0	16.1	16.3	8.9	11.7	20.2	18.2	21.4	22.3	19.4	21.2
	o	9.0	10.4	12.1	13.4	13.9	18.7	14.4	8.8	21.4	15.7	16.5	22.9	26.4
	ng	10.5	16.3	22.0	24.6	23.5	24.9	26.1	21.5	25.3	24.1	28.8	26.9	25.9
	m	12.0	16.6	20.5	20.7	22.5	19.2	24.9	27.3	13.7	18.2	22.5	24.8	27.5
	o	13.5	22.6	18.7	26.4	25.7	27.3	25.7	23.1	17.1	15.8	18.2	27.4	28.9
	𝔼	15.0	18.0	18.0	20.2	21.0	22.1	24.0	27.0	26.2	24.9	17.1	18.8	21.8

Fig. 2. Alignment matrix for a word "gashirakomori" (g/a/sh/i/r/a/k/o/m/o/r/i/). 𝕊 and 𝔼 respectively denote the start and end point.

Progress mode: If a user considers that y_i is not a correct phoneme sequence, but is better than y_{i-1}, she/he can continue the correction process by making a corrective utterance as "no, the right pronunciation is <*oov*>."

Return mode: If a user considers that y_i is worse than y_{i-1}, she/he can return y_i to y_{i-1} by an utterance as "back to the previous."

2.2. Word segment error locating

The proposed method allows a user to make a corrective utterance by repeating a part of the word. In comparison to correction using a whole word, the advantages of correction using a word segment can be considered as that the error locating in an output phoneme sequence becomes easier, and the mis-correction of the correct part of the output phoneme sequence can be prevented. However, to perform a word segment error locating, the system should be able to detect which part of the output phoneme sequence corresponds to the corrective phoneme sequence.

To resolve this problem, we use an open-begin-end version of the dynamic programming matching algorithm (OBE-DPM) [9] with a phoneme distance measure calculated from a phonetic confusion matrix. We build the phonetic confusion matrix using the ATR Japanese speech database C-set (a database for a large number of speakers (137 males and 137 females), including $142, 480$ speech samples with a total of $834, 521$ phonemes) [10].

The OBE-DPM is run between two phoneme sequences: a corrective phoneme sequence x and an output phoneme sequence y, to find a sub-sequence in y which is similar to x. This sub-sequence is treated as the target phoneme sequence that includes phoneme errors which need to be corrected. In the example shown in Fig. 2, a sub-sequence "sh i r a o n o" in y is obtained as the sub-sequence which includes phoneme error(s). The conflicting phoneme pairs of the sub-sequence and x are ('r', 'b'), ('ϕ', 'k'), ('n', 'm') and ('ng', 'ϕ'). These conflicting phoneme pairs are given to the process of the GPP-based phoneme correction.

2.3. GPP-based phoneme correction

Generalized posterior probability (GPP) has been used to verify the recognized entities at different levels, e.g., sub-word, word, and sentence [11]. In this study, we use GPP at the phoneme level.

An example of the GPP-based phoneme correction is shown in Fig. 3. It shows the output phoneme sequences y_{i-1} and y_i, and the corrective phoneme sequence x_i with the GPP values for each phoneme in them. The conflicting phonemes are indicated by squares. Among the conflicting phonemes, 'r' is not replaced with 'b', and 'n' is replaced by 'm' according to the GPP values. To deal with the insertion and deletion errors, we give a threshold of 0.5. In the example, y_{i-1} is judged to have a deletion error 'k', which is corrected by the threshold. In this example, y_i becomes a correct phoneme sequence.

Moreover, we assume that an error sequence should become different after a correction. The proposed method involves the algo-

y_{i-1}	g	a	sh	i	**r**	a	$\underline{\phi}$	o	$\underline{\text{n}}$	o	r	i	ϕ
GPP	0.93	0.66	0.61	0.72	**0.53**	0.92		0.74	0.34	0.66	0.95	0.99	
x_i	-	-	sh	i	$\underline{\text{b}}$	a	**k**	o	**m**	o	r	i	$\underline{\text{ng}}$
GPP	-	-	0.75	0.83	0.11	0.76	**0.55**	0.92	**0.86**	0.92	0.43	0.66	0.28
y_i	g	a	sh	i	**r**	a	**k**	o	**m**	o	r	i	~~ng~~
GPP	0.93	0.66	0.61	0.72	**0.53**	0.92	**0.55**	0.74	**0.86**	0.66	0.95	0.99	~~0.28~~

Fig. 3. An example of GPP based phoneme correction.

rithm which ensures that y_i is different from its previous versions $(y_0, \ldots, y_{i-1})$.

3.. EXPERIMENTS

3.1. Settings

We conducted experiment to evaluate the performance for the proposed method. We selected 25 Japanese words, including names of animals, plants, spheres, and Japanese and Chinese names from Wikipedia. These words were used in the experiment. The total number of phonemes was 305, and each word included 12.2 phonemes on average. Table 1 shows the Japanese pronunciations for the words.

18 native Japanese speakers participated in the experiment, including twelve males and six females. All subjects were given instructions, including an explanation of the user interface, and were permitted a trial use of the system.

In the experiment session, a subject sat on a chair 40cm from a SANKEN CS-3e directional microphone and taught a prepared word through speech interaction in Japanese. The output phoneme sequences were displayed in a monitor in front of the subject by katakana sequences. In the experiment, the maximum number M of corrective utterances was set to seven. In other words, at most a total of eight utterances could be made for each word.

In the experiment, we used ATRASR [12], which was developed by NICT, for phoneme recognition. The phoneme models were represented by HMM, which used mel-scale cepstrum coefficients and their delta parameters (25-dimensional) as features. The phoneme recognizer includes 26 Japanese phonemes.

To investigate the effectiveness of the word segment error locating, we experimented with the conditions of two kinds of correction: 1) "Segment-GPP" and 2) "Whole-GPP," each of which is described as follows:

Segment-GPP: Subjects were instructed to make a corrective utterance by repeating the whole word or part of the word, at the discretion of the subject. This is the proposed method.

Whole-GPP: Subjects were instructed to make a corrective utterance including the whole word.

We used phoneme accuracy (P%) and word accuracy (W%) for evaluation, each of which is defined as

$$\begin{aligned} P &= \frac{N_p - S - D - I}{N_p} \times 100 \\ W &= \frac{N_w - N_e}{N_w} \times 100 \end{aligned} \quad , \qquad (1)$$

where N_p and N_w denote the total number of phonemes and words used in the experiment, S, D and I respectively denote the total number of phonemes with substitution, insertion and deletion errors, and N_e denotes the total number of words which have misrecognized phonemes in each of them.

3.2. Baseline System

We employed a maximum likelihood (ML) based phoneme transcription method which was proposed by [4] as a baseline. In this method, each phoneme sequence in the N-best[*1] phoneme recognition result for each of speech samples $\{u_0, \ldots, u_i\}$ is applied to all of these speech samples, and the phoneme sequence with a highest average likelihood is treated as the output phoneme sequence of the word.

	Japanese pronunciation
1	n/a/m/i/h/a/r/i/n/e/z/u/m/i/
2	m/a/d/a/g/a/s/u/k/a/r/u/m/i/d/o/r/i/j/a/m/o/r/i/
3	k/u/r/o/s/u/t/e/n/a/g/a/z/a/r/u/
4	m/i/k/u/r/o/s/u/t/o/n/i/k/u/s/u/
5	k/i/k/u/g/a/sh/i/r/a/k/o/m/o/r/i/
6	m/i/s/e/s/u/k/u/m/i/k/o/
7	k/a/s/u/m/i/z/a/k/u/r/a/
8	t/o/k/i/w/a/m/a/ng/s/a/k/u/
9	b/u/t/a/ng/sh/i/r/o/m/a/ts/u/
10	k/i/b/a/n/a/k/a/t/a/k/u/r/i/
11	a/ng/d/o/r/o/m/e/d/a/s/e/u/ng/
12	k/a/m/i/n/o/k/e/z/a/b/e/t/a/s/e/
13	s/a/ng/g/u/r/e/z/a/
14	m/a/z/e/r/a/n/i/k/u/s/u/t/o/r/i/m/u/
15	r/i/zh/i/r/u/k/e/ng/t/a/u/r/u/s/u/
16	h/a/r/a/t/a/k/a/sh/i/
17	g/o/ng/s/u/ng/z/a/ng/
18	n/o/g/u/ch/i/h/i/d/e/j/o/
19	j/o/s/a/n/o/a/k/i/k/o/
20	b/a/o/z/u/ng/
21	a/zh/i/s/a/i/
22	t/a/n/u/k/i/
23	j/o/sh/i/o/
24	k/a/r/u/p/i/s/u/
25	a/k/u/e/r/i/a/s/u/

Table 1. The pronunciations of the words used in the experiment.

The speech samples recorded during the "Whole-GPP" experiment were used to evaluate the baseline system. In the "Whole-GPP" experiment, a correction process stopped when an output phoneme sequence became correct. Therefore, there were some words that do not have eight speech samples. After the experiment of "Whole-GPP," we collected speech samples to ensure that each word has eight speech samples. As a result, we obtained 2,800 speech samples (200 speech samples for one subject). These speech samples were used to evaluate the baseline system.

To evaluate the effect of the spoken interactive correction, we run the baseline system on both batch and interactive ways, each of which is described as follows:

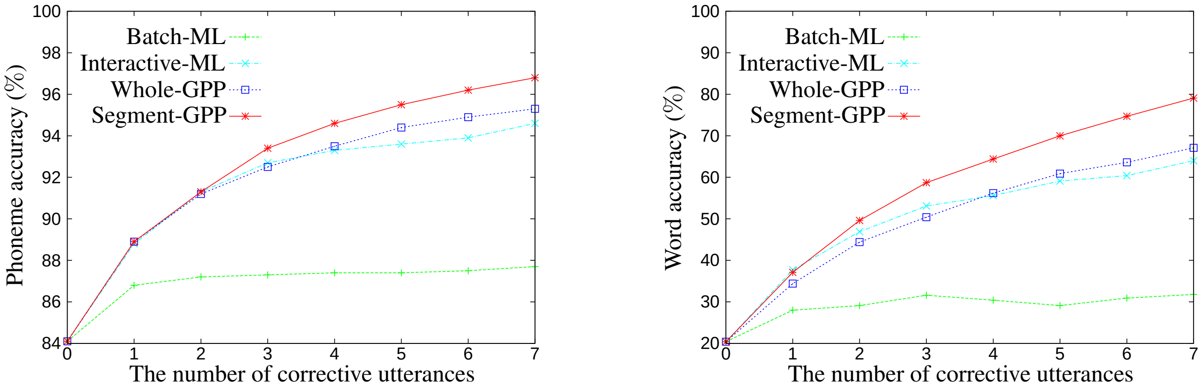
Batch-ML: The baseline system was run on a batch processing without any interactions with subjects. This is the same method proposed in [4].

Interactive-ML: The baseline system was run on an interactive way. We used the ML method in the proposed method to perform a correction for an output phoneme sequence. In the ML method, a corrective utterance with a word segment is not possible.

3.3. Results

The average phoneme and word accuracies obtained by 18 subjects are shown in Fig. 4 (a) and Fig. 4 (b), respectively. The horizontal axis represents the total number of corrective utterances ('0' represents the initial utterance). The performances for "Segment-GPP," "Whole-GPP," "Interactive-ML," and "Batch-ML" are shown in these figures. The average phoneme and word accuracies for the initial utterance were 84.1% and 20.4%, respectively. These values represent the performance of our

[*1] In the experiment, N was set to 50.



(a) Phoneme accuracies

(b) Word accuracies

Fig. 4. Phoneme and word accuracies change with the increase in the number of corrective utterances.

phoneme recognizer without any correction. For the proposed method (“Segment-GPP”), the accuracies improved significantly, to 96.8% and 79.1% at the seventh corrective utterance. On the other hand, in the baseline method (“Batch-ML”) the accuracies did not improve significantly with one or more corrective utterances, and were only 87.7% and 31.8% at the seventh corrective utterance.

The validity of the spoken interactive correction can be shown by comparing “Interactive-ML” and “Batch-ML.” In the figures, the accuracies for “Interactive-ML” increased directly with the increment of the corrective utterances. On the other hand, the accuracies for “Batch-ML” did not improve directly. At the seventh corrective utterance, “Interactive-ML” achieved 94.6% in phoneme accuracy and 64.0% in word accuracy, which were much higher than those achieved by “Batch-ML.”

The validity of the word segment error locating can be shown by comparing “Segment-GPP” and “Whole-GPP.” When the number of corrective utterances was greater than three, the performance of “Segment-GPP” became greater than that of “Whole-GPP.” However, in the first two corrective utterances, the phoneme accuracies for “Segment-GPP” and “Whole-GPP” were almost same. The reason is that many errors were corrected at once in the first two corrective utterances in “Whole-GPP,” while some of the correct parts of phoneme sequences were mis-corrected.

The validity of GPP-based phoneme correction could not be observed comparing “Whole-GPP” with “Interactive-ML.” The GPP-based phoneme correction, however, enabled the word segment error locating.

Moreover, the average numbers of corrective utterances spoken in the experiments were 3.68, 3.71, and 3.27 for “Segment-GPP,” “Whole-GPP,” and “Interactive-ML,” respectively. As far as the words which were correctly learned within seven corrective utterances, the average numbers of corrective utterance were 1.97, 2.20, and 2.28 for “Segment-GPP,” “Whole-GPP,” “Interactive-ML,” respectively. This means that by using the methods which were run on an interactive way, the correct phoneme sequences could be obtained within about two corrective utterances for the most words.

4.. CONCLUSION

This paper proposed a method that enables systems to learn new words with the correct phoneme sequences through speech interaction with users. The remarkable point of the method is that speech recognition errors are corrected by speech. The original features of the method are 1) interactive correction by speech, 2) error location by word segments, and 3) GPP-based phoneme correction. The experimental results clearly showed the validity of these three features. For practical applications, however, the number of corrective utterances needed to learn a word should be minimized. Future work includes a psychological evaluation and refinements to method.

5.. REFERENCES

[1] M. Dredze et al., “Contextual information improves OOV detection in speech,” in *Proc. NAACL*, 2010.

[2] B. H. Juang et al., “Automatic speech recognition – a brief history of the technology,” *Elsevier Encyclopedia of Language and Linguistics, Second Edition*, 2005.

[3] N. Jain et al., “Creating speaker-specific phonetic templates with a speaker-independent phonetic recognizer: Implications for voice dialing,” in *Proc. ICASSP*, 1996, pp. 881–884.

[4] K. Itou et al., “Estimation of transcription of unknown word from speech samples in word recognition,” *IEICE Trans. D-II (Japanese Edition)*, vol. J83-D-II, no. 11, pp. 2152–2159, 2000.

[5] C. V. Bael et al., “Automatic phonetic transcription of large speech corpora,” *Computer Speech and Language*, vol. 21, no. 4, pp. 652–668, 2007.

[6] R. Taguchi et al., “Learning lexicons from spoken utterances based on statistical model selection,” in *Proc. INTERSPEECH*, 2009, pp. 2731–2734.

[7] H. Holzapfel et al., “A dialogue approach to learning object descriptions and semantic categories,” *Robotics and Autonomous Systems*, vol. 56, pp. 1004–1013, 2008.

[8] G. Chung et al., “Automatic acquisition of names using speak and spell mode in spoken dialogue systems,” in *Proc. NAACL*, 2003, pp. 32–39.

[9] H. Sakoe, “Two-Level DP-matching — a dynamic programming based pattern matching algorithm for continuous speech recognition,” *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. ASSP-27, no. 6, pp. 588–595, 1979.

[10] A. Kurematsu et al., “ATR Japanese speech database as a tool of speech recognition and synthesis,” *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.

[11] F. K. Soong et al., “Generalized word posterior probability (gwpp) for measure reliability of recognized words,” in *Proc. Special Workshop in Maui*, 2004.

[12] S. Nakamura et al., “The ATR multilingual speech-to-speech translation system,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 365–376, 2006.