# Spoken Term Detection Using Japanese Language Posteriorgram for Zero-Resource Language

Satoru Mizuochi, Yuya Chiba, Takashi Nose, Akinori Ito
Graduate School of Engineering, Tohoku University

Sendai, Japan

{satoru.mizuochi.p3@dc, yuya@spcom.ecei, tnose@m, aito@spcom.ecei}.tohoku.ac.jp

*Abstract*—**In this paper, we research a spoken term detection method for the detection of terms in zero-resource languages. The method uses the posteriorgram of Japanese phonemes extracted by a phoneme classifier combined with the dynamic time warping method. The advantage of the method is that the posteriorgram is speaker-independent. Moreover, since phoneme classifiers are trained to discriminate acoustic differences between phonemes of a language, they may be able to express acoustic differences to some extent, even when applied to different languages. We exploited the deep neural network as a classifier and experimented with the spoken term detection from Japanese, English, and Kaqchikel speech. As a result, the method showed better detection performance compared with the method based on the direct comparison of mel frequency cepstrum coefficients. However, when the target language is English or Kaqchikel, the improvement is smaller than that obtained for Japanese.**

***Keywords-zero-resource language; spoken term detection; posteriorgram***

## I. INTRODUCTION

Currently, many minority languages are in danger of extinction. The number of languages in the world is estimated to be from 4000 to 6000. Most of them are considered to extinct during the century. Stopping the extinction of languages is difficult. Thus, there have been many attempts to keep records of languages while there are speakers of such languages. Among them, it is essential to save not only text languages but also spoken languages to create databases, and efforts to preserve many languages are being carried out. There are ambitious attempts such as the Human Language Project [1] that archives all the languages of the whole world, and there are attempts that utilize the corpus created in this way [2].

For such databases, it is desirable not only to record and catalog the spoken words but also to give a function to search the databases by words. Performing a search on speech is called “spoken document retrieval,” and in particular, the task of detecting a particular word among them is called "spoken term detection" (STD) [3]. There are several approaches to perform STD. One method is to transcribe the speech database to be searched by a speech recognizer and to perform the text retrieval. This method can perform the search quickly. To exploit this method, we need an accurate speech recognizer of a target language. It is generally difficult to realize highly accurate speech recognizer for minority languages. Although there are a few studies on the development of speech recognition systems for languages with low language resources such as minority languages [4], other methods do not recognize the speech database. This kind of methods uses a voice input as a search word, and search the database by signal processing method. Since the method of searching for a spoken keyword does not depend on languages, this method can be applied to any language regardless of the amount of language resources for system development.

We proposed a method that directly compares a voice query and speech in the database [5]. In [5], the proposed method showed better detection performance compared with the method based on the mel frequency cepstrum coefficients (MFCC). However, the performance was still low.

In this paper, we investigate a spoken term detection using the Japanese language for a zero-resource language. The method uses the cross-lingual phoneme posteriorgram, where a classifier of Japanese phonemes is applied to other languages. To extract the phoneme posteriorgram, we need to train the language-dependent feature extractor, which is difficult or impossible for languages with no language resources. Therefore, we extract the phoneme posteriorgram of a zero-resource language by applying Japanese phonemes.

## II. Spoken Term Detection Using Non-linear matching

### A. Overview

Detection of words form continuous speech originates in the 1980s. At that time, this task was called "word spotting." A nonlinear matching such as Dynamic Time Wrapping (DTW) between the spoken keyword and the speech in the database was executed for word spotting, which minimizes Euclidean distance between the voice features of the database and the search keyword [6]. However, this method has a drawback that distance between speech features depend on not only the difference of pronunciation but also the difference of the speakers. To reduce the influence of differences of feature vectors induced by the speaker difference, the feature vectors are converted into the speaker-independent features (such as posteriorgram), and then the distance between the speaker-independent feature is calculated [7]. In this case, we need to train the language-dependent feature extractor, which is difficult or impossible for languages with no language resources. There have been a few attempts to design a cross-lingual posteriorgram [8].

The method we research in this paper uses the posteriorgram extracted by the classifier of Japanese phonemes. The keyword is detected based on the result of DTW using the posteriorgram as a feature.

### B. Detection of words using DTW

The Dynamic Time Wrapping (DTW) is a method to find matching between the spoken input and the speech n the database. The basic DTW matches feature vectors of the input speech $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_I$ and the spoken keyword $\mathbf{Y} = \mathbf{y}_1, \ldots, \mathbf{y}_J$ as follows. First, let us consider the correspondence between two feature sequence $\Phi$ :

$$\Phi = (\phi_1, \ldots, \phi_N) \tag{1}$$

$$\phi_k = (i, j), \quad 1 \le i \le I, 1 \le j \le J \tag{2}$$

$$\phi_1 = (1,1), \quad \phi_N = (I, J) \tag{3}$$

Here, when we have $\phi_k = (i_k, j_k)$ and $\phi_{k'}$ where $k \le k'$ then $i_k \le i_{k'}$ and $j_k \le j_{k'}$.

Then the DTW finds the correspondence with minimum distance as follows.

$$\hat{\Phi} = \arg\min_{\Phi} D(\mathbf{X}, \mathbf{Y}, \Phi) \tag{4}$$

$$D(\mathbf{X}, \mathbf{Y}, \Phi) = \sum_{i=1}^{N} d\left(\mathbf{x}_{i_k}, \mathbf{y}_{j_k}\right) \tag{5}$$

Here $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between the two vectors. $\hat{\Phi}$ denotes the optimum correspondence between the two vector sequences, which is called "the optimum DP path." This optimization problem can be solved using dynamic programming, as follows.

$$d_{i,j} = \begin{cases} d(\mathbf{x}_i, \mathbf{y}_j) & 1 \le i \le I \text{ and } 1 \le j \le J \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

$$g(i, j) = d_{i,j} + \min \begin{cases} g(i-2, j-1) + d_{i-1,j} \\ g(i-1, j-1) \\ g(i-1, j-2) + d_{i,j-1} \end{cases} \tag{7}$$

$$D(\mathbf{X}, \mathbf{Y}, \Phi) = g(I, J) \tag{8}$$

Then $\hat{\Phi}$ is determined as the best matching path that is obtained by tracing the minimum decision in Eq. (7).

To detect a keyword from a longer speech database, we need to apply a different approach from the DTW mentioned above, since the original DTW is to match the vector sequences with almost similar lengths. Such an algorithm is called "the continuous DP matching" [6].

Let us assume that $I << J$ and

$$\mathbf{X}(i', i) \equiv \mathbf{x}_{i'}, \ldots, \mathbf{x}_i \tag{9}$$

Then the detection score of **Y** from **X** at position *i* is obtained as follows.

$$D(\mathbf{X}, \mathbf{Y}, i) = \min_{i', \Phi} D(\mathbf{X}(i', i), \mathbf{Y}, \Phi) \tag{10}$$

This score can be calculated in almost the same way as Eq. (7). The difference is that the start and endpoints of the sequence **Y** are not fixed to the start and endpoints of **X**. The score can be obtained by

$$D(\mathbf{X}, \mathbf{Y}, i) = g(i, J) \tag{11}$$

Fig. 1 shows an example of term detection. In this example, the spoken word "*nihon*" (Japan) is detected from the continuous speech spoken by the same speaker as the keyword speech. The curve in a black line shows the score, and the minima are the candidates of the detected keywords. The green line shows the threshold, and the minima larger than the threshold are not detected. The red dots show detected candidates.

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

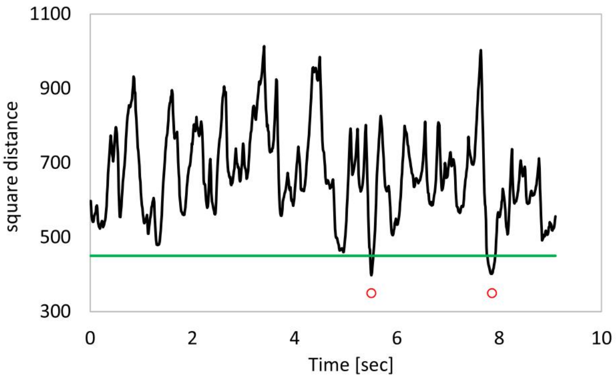


Figure 1. An example of STD using DTW

As reviewed in the introduction, information of the phoneme and the speaker is involved in a feature vector. Therefore, if speakers of the database and the keyword are different, the distance between feature vectors reflects not only the difference of pronunciation but also the individuality of the speech. The difference caused by the individuality is often larger than that from the difference of pronunciation, which is the main cause of degradation of the keyword detection performance.

### C. *The keyword detection based on phoneme posteriorgram and DTW*

The conventional methods exploit feature vectors that are independent of speakers and express only phonetic variation, such as the posteriorgram. Since the posteriorgram uses phonetic information, it inevitably depends on the language. However, since phoneme classifiers are trained to discriminate acoustic differences between phonemes of a language, they may be able to express acoustic differences to some extent, even when applied to different languages. Therefore, in this paper, we research the method which uses the posteriorgram obtained from a Japanese phoneme classifier for the target language.

Fig. 2 shows an outline of STD using the phoneme posteriorgram. In this method, we extract the acoustic feature from both a voice query and speech in the database. Here, the acoustic feature quantity is the MFCC of 12 dimensions and its parameter (24 dimensions in total). Next, the acoustic feature sequences are converted into the phoneme posteriorgram using a Japanese phoneme classifier. We use a phoneme classifier based on Deep Neural Network (DNN) for extracting the posteriorgram and use a Japanese speech corpus for training the phoneme classifier. After that, the word is detected by performing continuous DP matching using the posteriorgram obtained from a voice query and speech in the database.

In this paper, we first check the performance of STD when the target language is Japanese, and then evaluate the performance when the target language is English or Kaqchikel. Kaqchikel is a language of the Maya family spoken in Guatemala, and it is estimated that there are about 450,000 speakers. Kaqchikel gathers interest of linguists because of its "verb first" word order [10].

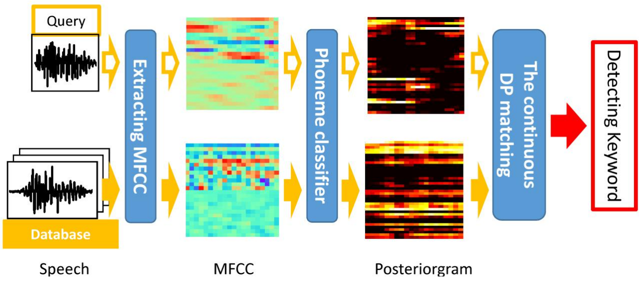


Figure 2. Outline of the research method

## III. EXPERIMENT

### A. *Training of the phoneme classifier*

To learn the phoneme classifier, we selected 4030 sentences spoken by 6 male speakers and 4 female speakers, as well as 1000 sentences spoken by 10 male and female speakers. The sentences were the phoneme-balanced sentences in the ATR database, a database of read Japanese speech.

Table I shows the parameter of DNN used for phoneme classifiers. A fully-connected multi-layer perceptron (MLP) was used as the DNN. The output units correspond to 38 classes of Japanese phonemes. We use MFCC segment as the input of DNN. We examined four conditions of the number of frames of a segment, as shown in Table I. We examined several numbers of hidden layers and the number of units. We only show the conditions with the highest phoneme accuracy rate in the table.

TABLE I. THE TRAINING PARAMETERS

| Length of input frames | 1 | 9 | 15 | 17 |
|---|---|---|---|---|
| Hidden layers | 3 | 7 | 4 | 7 |
| Units in a hidden layer | 300 | 2048 | 2048 | 2048 |
| Activation | ReLU | | | |
| Epoch | 10 | | | |
| Dropout | 0.5 | | | |

When the experiments of the phoneme classification were performed using 500 sentences spoken by 5 male and 5 female speakers not included in the training data, the phoneme accuracy was about 66 % when the input is 1 frame and about 78 % when the input is multiple frames.

### B. *Experimental conditions*

The Japanese evaluation data was 400 sentences randomly selected from newspaper article sentences in JNAS database. The voice queries were 7 words ("*kettei*," "*kotoshi*," "*seifu,*" "*chugoku*," "*nihon*," "*pasento*," and "*mondai"*) uttered by speakers not included in the evaluation data. 400 sentences were randomly selected from TIMIT database were used for English evaluation. We used "greasy," "oily," "suit," "wash," "water" and "year" uttered by speakers not included in the evaluation data as the voice queries. For Kaqchikel evaluation, we used Kaqchikel conversation voice recorded in a quiet environment (16 dialogues, 342 utterances, about 14minutes). From here, we detected four words "*achike*," "*matyöx*," "*peraj*" and "*richin*" uttered by speakers not included in the conversational speech. The voice queries were cut from continuous speech. Since the voice query is often an isolated utterance which includes silence before and after in actual systems, we inserted white noise whose length is equal with the frames spliced when the acoustic features are input to the phoneme classifier before and after the cut speech segments in this paper. The white noise's amplitude was 1, where the speech was quantized in 16 bits per sample.

In this paper, the evaluation is performed on an utterance basis. Therefore, we regarded the detection as correct when the query was found in an utterance that includes the query word. The detection results were evaluated using Mean Average Precision (MAP). Average Precision (AP) is the average value of the precision when the detection results are output from the top, and the correct answer is output. MAP is the mean value of the AP of each query. AP and MAP is obtained by

$$AP(q)=\frac{1}{c_q}\sum_{i=1}^{N}\delta_i P(q,i) \qquad (12)$$

$$MAP=\frac{1}{Q}\sum_{q=1}^{Q}AP(q) \qquad (13)$$

Here, $c_q$ is the number of correct utterances for query $q$, and $N$ is the number of the target utterances of detection. $\delta_i$ is the binary function, which is 1 when the $i$-th utterance of the detection results is correct and 0 when it was incorrect. $P(q,i)$ is precision when the $i$-th utterance of the detection results in query $q$, and $Q$ is the number of all queries.

### *C. Results*

Fig. 3 shows the result of the keyword detection of each language using the posteriorgram extracted by the phoneme classifiers trained in section III-A. The baseline in the figure is the result using the MFCC for calculation of the continuous DP score. Using the posteriorgram under any conditions, the performance in Japanese was improved compared with using the MFCC. This result shows that the speaker's characteristics are reduced by using the phoneme posteriorgram. From the result, we confirmed that the trained phoneme classifiers could extract the relevant posteriorgram of Japanese phonemes to some extent.

The result of English shows that the use of the posteriorgram improved the performance compared with the MFCC. However, the improvement was not as large as in Japanese.

The performance in Kaqchikel was improved compared with using the MFCC using the posteriorgram when the segment length was 9 or 17. The performance improvement was not as large as in Japanese, which was similar to the result of English.

The result in English and Kaqchikel suggested that the difference of the language influence the performance of speech detection using the posteriorgram.

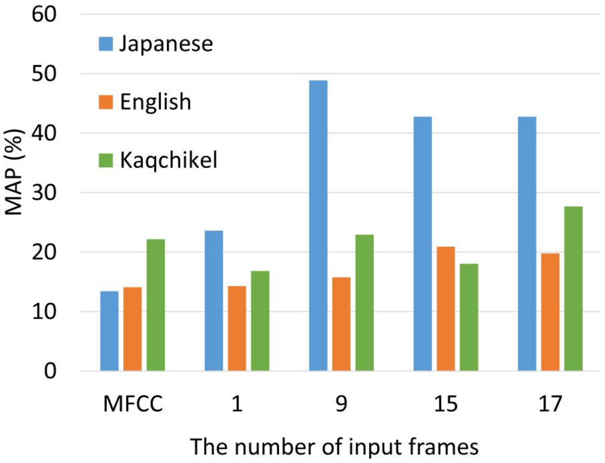
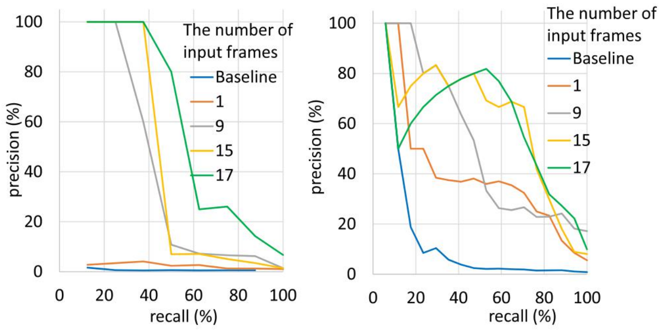


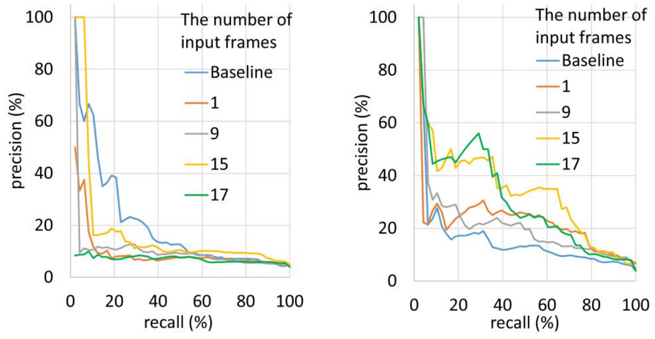
Figure 3. The detection result

Fig. 4 to 6 show examples of the recall-precision curves. In Japanese, the performance improved using the posteriorgram compared with MFCC in all conditions. In English or Kaqchikel, the performance improvement depended on the condition of the extractor and the word. The above results suggest that we should make further investigation of features that are robust of language difference.



(a) word "chugoku" (b) word "nihon"

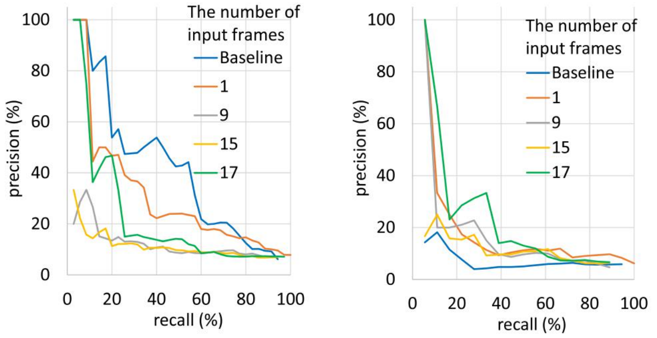
Figure 4. Recall-precision curve (Japanese)



(a) word "greasy'

(b) word "year"

Figure 5. Recall-precision curve (English)



(a) word "achike"

(b) word "richin"

Figure 6. Recall-precision curve (Kaqchikel)

## IV. Conclusion

In this paper, with the goal of word detection for zero-resource language, we researched STD using the posteriorgram extracted by a phoneme classifier trained in a language different from the target language. As a result of the experiments, performance of STD improved compared to the MFCC by using the posteriorgram for a different language. However, the improvement is smaller than when the language of the target and posteriorgram are the same. For future work, we will investigate the extraction of the multilingual posteriorgram and bottleneck feature.

## Acknowledgment

This work was supported by Research Platform for Yotta-scale Data Science and KAKENHI 19H05589.

## References

[1] S. Abney and S. Bird, "The human language project: Building a universal corpus of the world's languages," Proc. 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 88-97.

[2] T. McEnery, P. Baker and L. Burnard, "Corpus resources and minority language engineering," Proc. LREC, 2000.

[3] A. Mandel, K.R.P. Kumar, and P. Mitra, "Recent developments in spoken term detection: a survey," Int. J. of Speech Tech., vol.17, no.2, pp.183–198, 2014.

[4] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," Speech Communication, vol.56, pp.85–100, 2014.

[5] A. Ito and M. Koizumi, "Spoken term detection of zero-resource language using machine learning," Proc. ICIIT, pp.45-49, 2018.

[6] S. Nakagawa, "Connected spoken word recognition algorithms by constant time delay DP, O(n) DP and augmented continuous DP matching," Information Sciences, vol.33, no.1-2, pp.63-85, 1984.

[7] H. wang, T. Lee, C.C. Lenug, B. Ma, and H. Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," Proc. ICASSP, pp.8545-8549, 2013.

[8] R. Prabhavalker, K. Livescu, E. Fosler-Lussier, and J. Keshet, "Discriminative articulatory models for spoken term detection in low-resource conversational settings," Proc. ICASSP, pp.8287-8291, 2013.

[9] R.M. Brown, J.M. Maxwell and W.E. Little, "La ütz awäch?: Introduction to Kaqchikel Maya language," University of Texas Press, 2010.

[10] M. Koizumi, Y. Yasugi, K. Tamaoka, S. Kiyama, J.Kim, J.E.A. Sian, and L.P.O.G. Mätzer, "On the (non) universality of the preference for subject-object word order in sentence comprehension: a sentence processing study in Kaqchikel Maya," Language, vol.90, no.3, pp.722–736, 2014.