INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Recurrent Neural Network-based Phoneme Sequence Estimation using Multiple ASR Systems' Outputs for Spoken Term Detection

Naoki Sawada[1], *Hiromitsu Nishizaki*[2]

[1]The Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences,
[2]Graduate School of Interdisciplinary Research, Faculty of Engineering,
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi, 400-8511 JAPAN
sawada@alps-lab.org, hnishi@yamanashi.ac.jp

Abstract

This paper describes a novel correct phoneme sequence estimation method that uses a recurrent neural network (RNN)-based framework for spoken term detection (STD). In an automatic speech recognition (ASR)-based STD framework, ASR performance (word or subword error rate) affects STD performance. Therefore, it is important to reduce ASR errors to obtain good STD results. In this study, we use an RNN-based phoneme estimator, which estimates a correct phoneme sequence of an utterance from some sorts of phoneme-based transcriptions produced by multiple ASR systems in post-processing, to reduce phoneme errors. With two types of test speech corpora, the proposed phoneme estimator obtained phoneme-based N-best transcriptions with fewer phoneme recognition errors than the N-best transcriptions from the best ASR system we prepared. In addition, the STD system with the RNN-based phoneme estimator drastically improved STD performance with two test collections for STD compared to our previously proposed STD system with a conditional random fields-based phoneme estimator.

Index Terms: correct phoneme estimation, post-processing, recurrent neural network, spoken term detection

1. Introduction

Spoken term detection (STD), a speech data retrieval technology, is designed to determine whether a given utterance includes a query term consisting of a word or phrase. STD research has become popular topic in spoken document processing studies, and the number of STD research reports has increased following the 2006 STD evaluation organized by the National Institute of Standards and Technology [1].

The difficulty in STD lies in the search for terms under a vocabulary-free framework because search terms are not known prior to a large vocabulary continuous speech recognition (LVCSR) system. In the past, most STD studies have focused on the out-of-vocabulary (OOV) problem. For example, many papers have proposed subword (syllable or phoneme)-based STD approaches [2], which were very robust for OOV queries.

Another difficulty is that STD is weak against speech recognition errors. For example, the speech recognition performance (word or subword error rates) of target speech affects STD performance in a matching process between a subword sequence of a query term and a subword-based transcription of the target speech under a subword-based STD framework. Therefore, to obtain good STD results, improving automatic speech recognition (ASR) performance for target speech is also important. However, it is nearly impossible to completely remove ASR errors even when state-of-the-art ASR technologies such as deep learning-based acoustic modeling are used. Therefore, an STD technique that is robust against ASR errors is required. For example, a lattice-based STD approach [3, 4] has been proposed. Lattice representation of an ASR result of an utterance has richer word (or subword) sequence information than a single word (or subword) sequence output. Therefore, a query term can be flexibly matched against a lattice.

This paper proposes a novel framework that estimates correct phoneme sequences from multiple ASR system outputs of search-targeted speech using a recurrent neural network (RNN) framework for an STD task. We use a long short-time memory (LSTM [5])-based correct phoneme sequence estimator to improve a subword-based transcription of search-targeted speech.

An LSTM-based network model is widely used in various ASR tasks, such as language modeling [6, 7] and acoustic modeling [8, 9], and it is known to perform extremely well for those tasks. Therefore, in this study, we apply an LSTM-based framework to correct phoneme sequence estimation in an ASR post-processing phase. We show that the error-corrected phoneme-based transcriptions produced by the LSTM-based estimator improve the STD performance on STD test collections.

The basic idea of this study is to predict a correct phoneme using the phoneme history (sequence of phonemes) with an RNN. The proposed LSTM-based framework is the same as a previously proposed RNN-based language model (LM) [6]. However, differing from previously proposed RNN-based language modeling frameworks, this study examines phoneme-based ASR error diversity in various ASR system outputs for phoneme estimation. In other words, the proposed LSTM-based phoneme estimator trains phoneme error patterns in the history, i.e., phoneme sequences produced by multiple ASR systems and estimates a correct phoneme using the phoneme error pattern history.

Because our approach uses only phoneme-based transcriptions of speech data created by ASR system(s); thus, it can be applied to phoneme-based transcriptions from any ASR systems. This is an advantage of the proposed framework. In this study, we use 10 types of ASR systems with different acoustic models (AMs) and LMs.

The proposed approach is also similar to the error correction process in the Recognizer Output Voting Error Reduction (ROVER) method proposed by Fiscus [10]. The ROVER

http://dx.doi.org/10.21437/Interspeech.2016-337

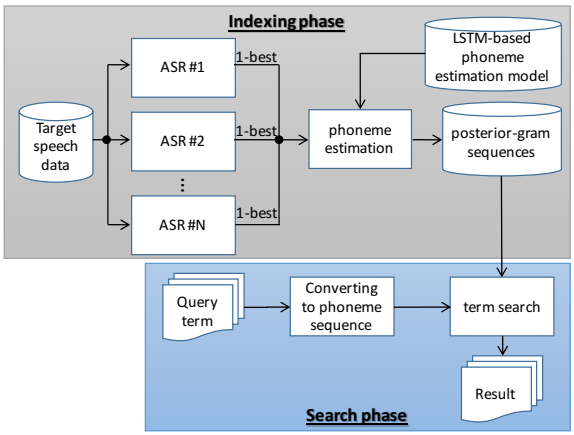

Figure 1: STD flow with phoneme estimation.

method can reduce ASR errors by combining multiple ASR system outputs using a voting process. We use a machine learning (deep learning) technique with the ROVER framework's error correction process to generate more accurate N-best transcriptions of search-targeted speech.

In this paper, we show that the LSTM-based phoneme estimator can obtain phoneme-based N-best transcriptions with fewer phoneme recognition errors compared to the N-best transcriptions from the best ASR system. In addition, we show that, compared to our previously proposed STD system with a conditional random field (CRF)-based triphone estimator [11], the proposed LSTM-based phoneme estimator system improves STD performance on two STD test collections. The experimental results show that, on the two test collections, the STD system with the phoneme estimator achieved an improvement of 0.088 and 0.099 points (i.e., 11.6% and 21.5%, respectively) in mean average precision (MAP) compared to the CRF-based triphone estimator.

2. LSTM-based Phoneme Sequence Estimation

Figure 1 illustrates the STD process with LSTM-based correct phoneme sequence estimation. First, search-targeted speech data are speech-recognized by N ASR systems[1]. Then, an LSTM-based phoneme estimator outputs posterior-gram sequences. In this study, such sequences are used as rich representations of phoneme-based transcriptions. Next, an STD engine searches a query term for the posterior-gram representation of the target speech data.

2.1. LSTM-based phoneme estimator

The LSTM-based phoneme estimator process is shown in Figure 2. In Figure 2, V_n is the n-th input vector and p_n is the n-th estimated phoneme, which is determined by the softmax function in the output layer. Here the LSTM-based phoneme estimator has one LSTM hidden layer and four fully-connected neural network layers. The number of nodes in each hidden layer is 512. We use a rectified linear unit [12] as an activation function and stochastic gradient descent to update parameters. The number of nodes in the output layer is 35, which is equal to the number of phoneme classes.

Figure 3 shows an example vector representation of input

[1]We use 10 ASR systems in this paper. However, the proposed method is not bound by the number of ASR systems.

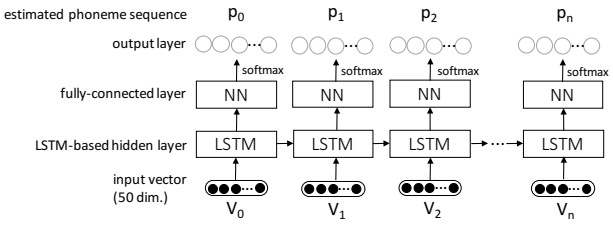

Figure 2: LSTM-based phoneme estimator.

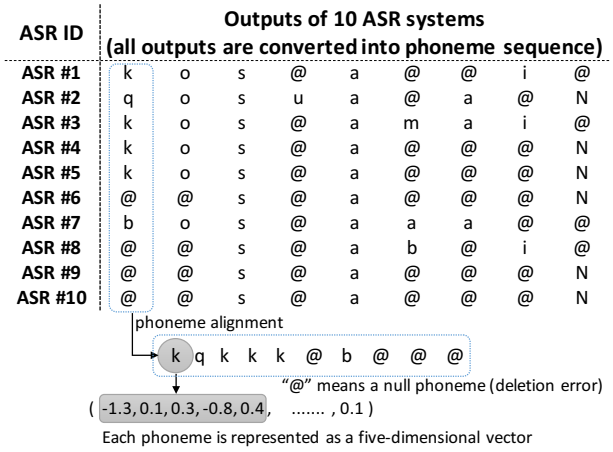

Figure 3: Vector representation of an input vector.

vectors. Speech data were recognized by the 10 ASR systems to yield 10 hypotheses. Then, they were converted into phoneme sequences. Next, we obtain a "phoneme-alignment" sequence using a dynamic programming (DP) scheme [10]. A phoneme-alignment is input to the LSTM-based phoneme estimator.

Each phoneme-alignment has 10 phonemes, including null (phoneme deletion). When a word or a phoneme is converted to a fixed dimensional vector, typically a 1-of-N representation [13] is used. In this case, a phoneme can be represented as a 35-dimensional vector because we deal with 35 types of phonemes. However, we want the LSTM to train phoneme-to-phoneme confusion patterns. Therefore, we convert aligned phonemes to a vector by considering the similarity between phonemes based on Bhattacharyya distance (BD) [14]. The BD between phoneme p and q is calculated by the monophone-based Gaussian mixture models of p and q. For the distance matrix between all phonemes, we apply principal component analysis to the matrix. Finally, we obtain a five-dimensional vector for each phoneme by using up to five principal components. Therefore, the number of dimensions of the input vector is 50. A deletion error (@) is replaced by the subsequent phoneme.

2.2. STD engine

Figure 4 shows an example of the term search process for a query consisting of seven phonemes for a posterior-gram sequence of target speech data. The search process is very simple, i.e., DP matching between a phoneme sequence of a query and a posterior-gram sequence.

In this example, the detection probability of the query is 0.6. The search engine simultaneously calculates the maximum probability of the query-detected region using the best probability of each posterior-gram. In this case, the maximum probability is 0.7. The final STD score for the query is 0.85, which is obtained by dividing the detection probability by the maximum

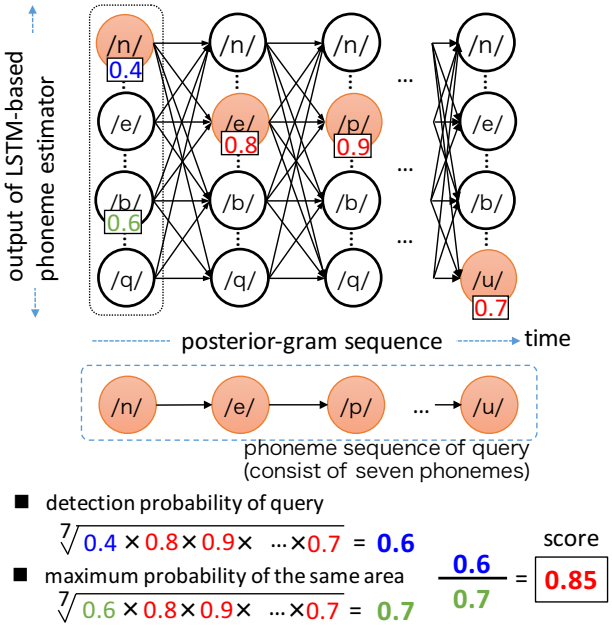

Figure 4: Calculation of a term detection score for a posterior-gram sequence.

probability in the same region. STD performance for a query set can be illustrated using a recall-precision curve, which is plotted by changing a threshold.

3. CRF-based Triphone Estimation

Previously, we proposed an STD method that used CRFs to directly detect a triphone in an utterance that is a part of a search targeted speech [11]. All words can be decomposed into a phoneme sequence. Therefore, we could use a triphone detector for word detection. In addition, context information is very useful. We created triphone detectors in consideration of phoneme-to-phoneme confusion error patterns in the CRF framework. The details of the CRF-based STD procedure is explained in the paper [11].

In this paper, we compare the CRF-based approach with the LSTM-based approach for an STD task.

4. Experiment

4.1. Experimental setup

4.1.1. Test collections

We used two types of STD test collections to verify the proposed method. One is the OOV subset of the Japanese test collection for STD [15] ("CSJ-OOV set"). This test collection targets speech data from 177 lectures (39 hours) in the Corpus of Spontaneous Japanese (CSJ) [16]. The number of utterances is 53,892. The OOV subset contains a total of 50 terms, which were spoken 233 times in the 177 lectures. The other test collection is the moderate-size task from NTCIR-10 SpokenDoc-2 [17], which contains speech data from 104 oral presentations (28.6 hours) from the first to sixth annual SDPWS ("SDPWS set"). In the SDPWS set, the number of query terms is 100, of which 47 are INV queries included in the ASR dictionary of the word-based trigram model and 53 are OOV. The occurrences of INV and OOV in the query set are 444 and 456, respectively.

4.1.2. ASR systems

As shown in Figure 3, the speech data were recognized by the 10 ASRs. Julius ver. 4.1.3 [18], an open source decoder for

Table 1: Phoneme correct rates of the N-best transcriptions on the CSJ speeches [%].

N-best	1-best	2-best	3-best	4-best	5-best
The best ASR	91.4	92.4	92.9	93.2	93.3
LSTM (10 ASRs)	**92.8**	**96.5**	**97.5**	**98.0**	**98.4**
LSTM (10-best)	91.5	94.5	95.5	96.1	96.5

Table 2: Phoneme correct rates of the N-best transcriptions on the SDPWS speeches [%].

N-best	1-best	2-best	3-best	4-best	5-best
The best ASR	83.5	84.6	85.0	85.3	85.5
LSTM (10 ASRs)	**86.9**	**91.7**	**93.5**	**94.8**	**95.8**
LSTM (10-best)	83.7	87.6	89.6	90.9	92.0

LVCSR, was used in all systems. We prepared two types of AMs and five types of LMs. The AMs are triphone-based (Tri.) and syllable-based hidden Markov models (HMMs) (Syl.) with both types of HMMs trained from the spoken lectures except for the 177 lecture speeches in the CSJ. All the LMs are word- and character-based trigrams as follows:

WBC: Word-based trigram in which words are represented by a mix of Chinese characters, Japanese Hiragana, and Katakana.

WBH: Word-based trigram in which all words are represented by only Japanese Hiragana. Words comprising Chinese characters and Katakana are converted into Hiragana sequences.

CB: Character-based trigram in which all characters are represented by Hiragana.

BM: Character sequence-based trigram in which the unit of language modeling is two Hiragana characters.

None: No LM is used. Speech recognition without any LM is equivalent to phoneme (or syllable) recognition.

Each model was trained from the many transcriptions in the CSJ under the open speech data [19].

Finally, 10 combinations, consisting of two AMs and five LMs, were formed.

4.1.3. Training set for models

Both the LSTM-based phoneme estimator and the CRF-based triphone estimator were trained with the training features created from the ASR transcriptions of the 2,525 lecture speeches in CSJ[2]. Note that these differed from the 177 speeches in the CSJ-OOV test collection.

4.1.4. Evaluation metrics

We evaluated the proposed method with two tasks, a correct phoneme estimation task and an STD task. We calculated phoneme correct rates as the evaluation metric for the phoneme estimation task.

The evaluation metrics for STD included recall, precision, F-measure of the optimal point on a recall-precision curve, and mean average precision (MAP) values [20]. These measures are officially used in the test collections.

[2]All AMs and LMs were trained under open condition for the target speech data.

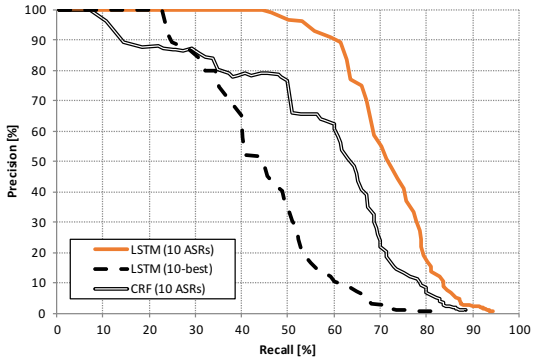

Figure 5: Recall-precision curves for each STD approach on the CSJ-OOV set.

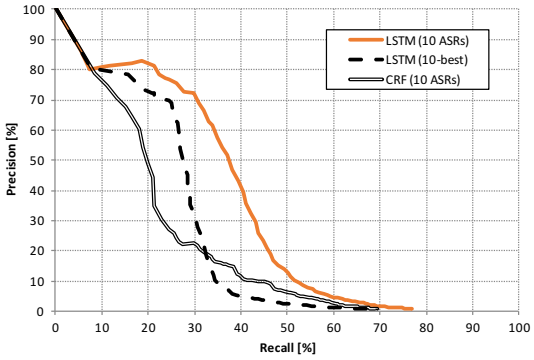

Figure 6: Recall-precision curves for each STD approach the SDPWS set.

Table 3: Max. F-measure and MAP values for each STD approach on the CSJ-OOV set.

Systems	max. F-measure [%]	MAP
LSTM (10 ASRs)	**72.8**	**0.847**
LSTM (10-best)	49.5	0.584
CRF (10 ASRs) [11]	61.1	0.759

Table 4: Max. F-measure and MAP values for each STD approach on the SDPWS set.

Systems	max. F-measure [%]	MAP
LSTM (10 ASRs)	**43.7**	**0.559**
LSTM (10-best)	37.0	0.390
CRF (10 ASRs) [11]	28.6	0.460

4.2. Phoneme estimation task

Tables 1 and 2 show the phoneme correct rates of the N-best transcriptions with the two test collections. The LM and AM in the best ASR system were "WBC" and "Tri.", respectively.

In this paper, the LSTM-based phoneme estimators were trained with two feature sets, one set was made of 10 ASR systems' 1-best outputs ("LSTM (10 ASRs)") and the other was made of the 10-best outputs of the best ASR system ("LSTM (10-best)").

As can be seen in Tables 1 and 2, "LSTM (10 ASRs)" obtained the best results for both the test collections. The LSTM-based estimator was trained with the transcriptions of the CSJ speeches. Despite that, as can be seen in Table 2, the LSTM-based estimator outperformed the best ASR system with the SDPWS speech data, which differs from the training speech corpus.

Furthermore, the feature set created by the 10 ASR systems generated a better phoneme estimator than the 10-best outputs from the best ASR systems. This indicates that the transcriptions from the different ASR systems were effective training data for this task. However, LSTM (10-best), which was trained with the 10-best hypothesis from the best ASR system, also improved the N-best hypothesis. In other words, the LSTM-based phoneme estimation worked well for the single ASR system output.

4.3. STD performance

Figures 5 and 6 show the recall-precision curves for the three STD frameworks with the CSJ-OOV set and the SDPWS set, respectively. Tables 3 and 4 show the F-measure values for the maximum points of the curves and MAP values on the test collections.

First, we compare the two transcriptions used to create the training features of the LSTM-based phoneme estimator. As shown in the abovementioned figures and tables, the LSTM trained with the multiple ASR system output significantly outperforms the LSTM trained using the 10-best transcriptions of the single (best) ASR system for all evaluation metrics with both test collections. These results demonstrate the effectiveness of the training feature set created using the multiple ASR systems outputs. The transcriptions from the multiple ASR systems have more varied phoneme-to-phoneme error confusion patterns than the N-best output of the single ASR system. These confusion patterns help machine learning-based error estimation.

Next, we compare the LSTM-based approach ("LSTM (10 ASRs)") with the CRF-based approach ("CRF (10 ASRs)"). Both the STD systems were trained with the feature set based on the transcriptions of the 10 ASR systems. These figures and tables indicate that the LSTM-based estimator has better phoneme estimation ability. In particular, the LSTM worked robustly on the SDPWS set, i.e., the different speech corpus from the CSJ used to train the phoneme estimation models, because the CRF-based approach was outperformed by the "LSTM (10-best)" relative to F-measure.

5. Conclusion

In this study, we proposed a novel LSTM-based correct phoneme sequence estimator for STD tasks. The proposed LSTM-based phoneme estimator, which was trained using a feature set based on multiple ASR system outputs, could generate more accurate phoneme-based transcriptions in the post-processing phase of ASR.

The LSTM-based phoneme estimator was evaluated with two tasks, i.e., correct phoneme estimation and STD. The experimental results of the phoneme estimation task show that the proposed LSTM-based estimator could output more accurate N-best transcriptions than the best ASR system. The STD system with the LSTM drastically improved STD performance with the two test collections for STD compared to our previously proposed CRF-based STD system.

In the future, we will use varieties of LMs, AMs, and other ASR systems, such as the Kaldi ASR toolkit [21], to prove the effectiveness of the proposed approach.

6. Acknowledgements

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Grant Number 26282049 and Grant-in-Aid for Scientific Research (C) Grants Number 15K00254.

7. References

[1] NIST, "The spoken term detection (STD) 2006 evaluation plan," 2006, http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf.

[2] S. Meng, J. Shao, R. P. Yu, J. Liu, and F. Seide, "Addressing the out-of-vocabulary problem for large-scale Chinese spoken term detection," in *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH 2008)*, pp. 2146–2149, 2008.

[3] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, no. 8, pp. 2338–2347, 2011.

[4] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowski, and O. Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in low resource languages," in *Proceedings of the 2013 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2013)*, pp. 8560–8564, 2013.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Time Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, pp. 1045–1048, 2010.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH 2012)*, pp. 194–197, 2012.

[8] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014)*, pp. 338–342, 2014.

[9] A. W. Senior, H. Sak, F. de Chaumont Quitry, T. N. Sainath, and K. Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNS," in *Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2015)*, pp. 604–609, 2015.

[10] J. G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition adn Understanding (ASRU'97)*, pp. 347–354, 1997.

[11] N. Sawada, S. Natori, and H. Nishizaki, "Re-Ranking of Spoken Term Detections Using CRF-based Triphone Detection Models," in *Proceedings of the 6th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2014)*, pp. 1–4, 2014.

[12] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the 14th International Conference on Articial Intelligence and Statistics (AISTATS 2011)*, pp. 315–323, 2011.

[13] H. Schwenk and J.-L. Gauvai, "Training Neural Network Language Models on Very Large Corpora," in *Proceedings of HLT/EMNLP 2005*, pp. 201–208, 2005.

[14] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," in *Proceedings of the fourth International Conference on Spoken Lanugage Processing (ICSLP'96)*, pp. 2005–2008, 1996.

[15] Y. Itoh, H. Nishizaki, X. Hu, H. Nanjo, T. Akiba, T. Kawahara, S. Nakagawa, T. Matsui, Y. Yamashita, and K. Aikawa, "Constructing Japanese test collections for spoken term detection," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, pp. 677–680, 2010.

[16] K. Maekawa, "Corpus of Spontaneous Japanese: Its design and evaluation," in *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR 2003)*, pp. 7–12, 2003.

[17] T. Akiba, H. Nishizaki, K. Aikawa, X. Hu, Y. Itoh, T. Kawahara, S. Nakagawa, H. Nanjo, and Y. Yamanashita, "Overview of the NTCIR-10 SpokenDoc-2 Task," in *Proceedings of the 10th NTCIR Conference*, pp.573–587, 2013.

[18] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine Julius," in *Proceedings of the 1st Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2009)*, pp. 131–137, 2009.

[19] H. Nishizaki, T. Akiba, K. Aikawa, T. Kawahara, and T. Matsui, "Evaluation Framework Design of Spoken Term Detection Study at the NTCIR-9 IR for Spoken Documents Task," *Journal of Natural Language Processing*, vol. 19, no. 4, pp. 329–350, 2012.

[20] T. Akiba, H. Nishizaki, K. Aikawa, T. Kawahara, and T. Matsui, "Overview of the IR for Spoken Documents Task in NTCIR-9 workshop," in *Proceedings of the 9th NTCIR Workshop Meeting*, pp. 223–235, 2011.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU 2011)*, 2011.