Evaluation of DNN-based Phoneme Estimation Approach on the NTCIR-12 SpokenQuery&Doc-2 SQ-STD Subtask

Naoki Sawada
University of Yamanashi
4-3-11 Takeda, Kofu-shi,
Yamanashi, 400-8511, Japan
sawada@alps-lab.org

Hiromitsu Nishizaki
University of Yamanashi
4-3-11 Takeda, Kofu-shi,
Yamanashi, 400-8511, Japan
hnishi@yamanashi.ac.jp

ABSTRACT

This paper proposes a correct phoneme sequence estimation method using a deep neural network (DNN)-based framework for spoken term detection (STD). We use a DNN architecture as a correct phoneme estimator. The DNN-based estimator estimates a correct phoneme sequence of an utterance from some sorts of phoneme-based transcriptions produced by multiple ASR systems in post-processing, for reducing phoneme errors. In the experimental evaluation on the NTCIR-12 SpokenQuery&Doc-2 SQ-STD test collection, our proposed approach defeated the baseline system prepared by the task organizers. However, our approach could not outperform our DTW-based STD method we previously proposed.

Team Name

ALPS

Subtasks

SQ-STD (Japanese)

Keywords

NTCIR-12, spoken term detection, deep neural network, correct phoneme estimation

1. INTRODUCTION

Spoken term detection (STD) is designed to determine whether or not a given utterance includes a query term consisting of a word or phrase. STD research has become a hot topic in the spoken document processing research field, and the number of STD research reports is increasing in the wake of the 2006 STD evaluation organized by National Institute of Standards and Technology [15].

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. Many studies tackling STD have already been proposed [18, 13]. In the past, most STD studies focused on out-of-vocabulary (OOV) and speech recognition error problems. For example, STD techniques using subword (syllable or phoneme)-based lattices or confusion networks (CN) have been proposed [13]. In recent works, we also proposed a CN-based indexing and a dynamic time warping (DTW)-based search engine [14]. The CN-based index, which we call "Phoneme Transition Network (PTN)-formed index [14]," was made of 10 types of transcriptions generated by the 10 different automatic speech recognition (ASR) systems, including an LVCSR system and a phoneme recognition system. We have shown that our proposed method could outperform other STD technologies that participated in the ninth National institute of informatics Testbeds and Community for Information access Research (NTCIR-9) project STD evaluation framework [3]. A DTW-based matching between a subword sequence of a query term and a transcription of speech is weak for speech recognition errors. Therefore, the STD performance of the DTW-based technique depends on the accuracy of subword-based transcriptions.

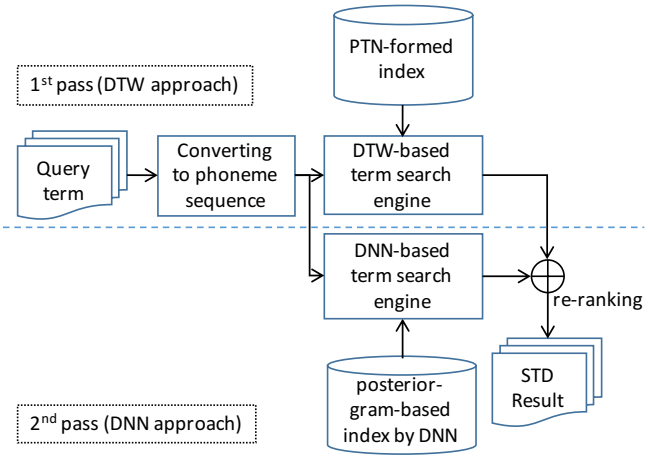

Figure 1: Overview of the two-pass STD framework using CRF-based triphone detection modeling.

Our DTW-based approach using a PTN-formed index for STD was very robust for ASR errors. However, this approach output many false detections because the structure of PTN was complex [14]. These false detections degraded the STD performance. In this paper, we focus on controlling false detections in a second-pass stage using a machine learning approach.

Figure 1 shows our STD framework. We explore a correct phoneme estimation approach using a deep neural network (DNN)-based framework for making an index for term detection in the second-pass stage on the NTCIR-12 SpokenQuery &Doc-2 SQ-STD test collection. A DNN-based phoneme estimator makes a posterior-gram-based index from phoneme-based transcriptions of target speeches outputted by multi-

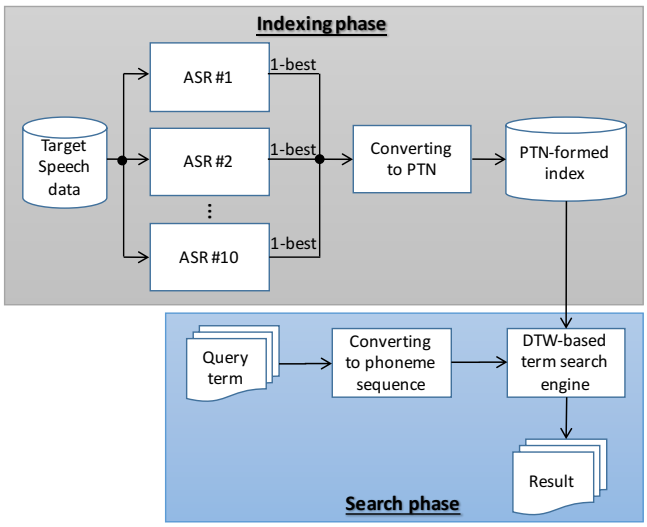

Figure 2: Overview of the first-pass stage using DTW-based matching.

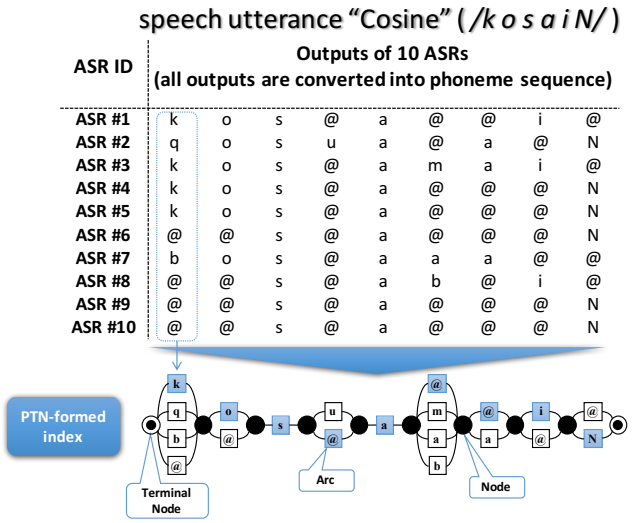

Figure 3: Generation of a PTN-formed index by performing alignment using DP and converting to a PTN.

ple ASR systems.

A DNN-based phoneme estimator is trained by using features generated from 10 types of phoneme-based transcriptions. This approach is sensible because the features for DNN modeling are prepared for making a PTN-formed index, which is also derived from 10 types of transcriptions, used in the first-pass of the entire STD framework.

2. DTW-BASED APPROACH

DTW-based STD using a PTN-formed index is performed in the first-pass stage on the entire STD framework, as in the baseline approach. Figure 2 shows an overview of the baseline method. In the indexing phase, speech data is processed using ASR, and the recognition outputs (words or sub-word sequences) are converted into the PTN-formed index for STD. Figure 3 shows an example of the development of a PTN-formed index for the speech "*cosine*" (Japanese pronunciation is `/k o s a i N/`) by aligning 10 phoneme sequences from the best hypothesis of all the ASR systems. The speech was recognized by the 10 ASR systems to yield 10 hypotheses, which were then converted into phoneme sequences. Next, we obtained "aligned sequences" using the same Dynamic Programming (DP) scheme, as described in [6]. Finally, a PTN was obtained by converting the aligned sequences. The term "@" in Fig. 3 indicates a null transition.

In the search phase, the word-formed query is converted into a phoneme sequence. Then, the phoneme-formed query is input to the term search engine. The term search engine searches for the query term from the index at the phoneme level using the DTW framework. Unlike the combination techniques of multiple STD systems, described in [1], the baseline system combines the transcriptions produced by multiple ASR systems.

Figure 4 shows an example of the DTW framework for the search term "*cosine*" (`/ k o s a i N /`) and the PTN-formed index. The PTN contains multiple arcs between adjoining node pairs. These arcs are compared to one of the phoneme labels of a query term.

We used edit distance as cost on the DTW paths. The details of the DTW matching process are discussed in our previous study [7, 14]. We calculate distance between a query and a PTN based on Bhattacharyya distance (BD) [12]. The BD between phoneme p and q is calculated by the monophone-based Gaussian mixure models (GMMs) of p and q.

$$BD(p,q) = \frac{1}{8}(\mu_p - \mu_q)^T \Sigma^{-1}(\mu_p - \mu_q) + \frac{1}{2}\ln\left(\frac{|\Sigma|}{\sqrt{|\Sigma_p||\Sigma_q|}}\right) \quad (1)$$

The costs for substitution, insertion and deletion errors were commonly set to 1.0 when the number of phonemes consisting of a query term was N or larger than N. On the other hand, each cost was commonly set to 1.5 when the number of phonemes was less than N to avoid false term detections in query terms, having less number of phonemes. This cost (=1.5) was optimized using a development query set. The total DTW cost $D(i,j)$ at the grid point (i,j) ($i = \{0,...,I\}$, $j = \{0,...,J\}$, where I and J are the number of the set of arcs in the index and query term, respectively) on the DTW lattice was calculated by the following equations:

$$D(i,j) = \min \begin{cases} D(i,j-1) + Del(i) \\ D(i-1,j) + Null(i) \\ D(i-1,j-1) + Match(i,j) + Vot(i,j) \end{cases} \quad (2)$$

$$Match(i,j) = \begin{cases} 0.0 : Query(j) \in PTN(i) \\ minBD : Query(j) \notin PTN(i) \end{cases} \quad (3)$$

$$Vot(i,j) = \begin{cases} \alpha \div (Voting(p) + \Sigma_q(BD(q) \\ \qquad \times BDVoting(q))) \\ \qquad : \exists p \in PTN(i), p = Query(j), \\ \qquad q \neq Query(j) \\ \alpha : Query(j) \notin PTN(i) \end{cases} \quad (4)$$

$$Del(i) = min \begin{cases} minBD(p,q) : \exists p \in PTN(i-1), \\ \qquad q = Query(j) \\ minBD(p,q) : \exists p \in PTN(i), \\ \qquad q = Query(j) \end{cases} \quad (5)$$

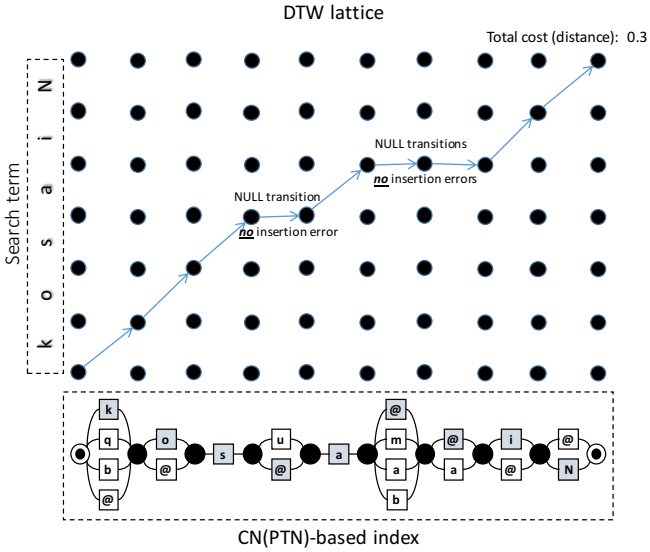

Figure 4: Example of term search on a network-formed index.

$$Null(i) = min \begin{cases} NullVot(i) : Null \in PTN(i) \\ minBD(p,q) : \exists p \in PTN(i-1), \\ \qquad q \in PTN(i) \\ minBD(p,q) : \exists p \in PTN(i), \\ \qquad q \in PTN(i+1) \end{cases} \quad (6)$$

$$NullVot(i) = \beta \div Voting(Null) \quad (7)$$

where $PTN(i)$ is the set of phoneme labels of the arcs at the i-th node in the PTN, and $Query(j)$ indicates the j-th phoneme label in the query term.

Equation (3) and Eq. (4) are related to the cost calculation for a substitution error. $minBD$ in Eq. (3) is the smallest BD between j and any phoneme in $PTN(i)$. $Vot(i,j)$ is a confidence paremeter for the matching between $PTN(i)$ and $Query(j)$. "$Voting(p)$" is the number of ASR systems outputting the same phoneme p at the same arc. More value of $Voting(p)$ makes reliability of phoneme p better. $BD(q)$ is the BD between $Query(j)$ and phoneme q, which is not correspond to $Query(j)$. $BDVoting(q)$ also means the number of ASR systems outputting the phoneme q. The cost for a deletion error is calculated based on Eq. (5). $minBD(p,q)$ is BD between phoneme p and q. Equation (6) and Eq. (7) are used for the cost calculation for an insertion error. We allow a null transition between two nodes in the PTN-formed index with the cost $NullVot(i)$ defined in Eq. (7). α and β are hyper parameters and set to 0.5 and 0.45, respectively. They were optimized using the development set. The appropriate null cost achieves increasing term detection and decreasing false detections.

In advance searches for the query term, the term detection engine initializes $D(i,0) = 0$ (both the end-points are free), and then, it calculates $D(i,j)$ using Eq.(2) ($i = \{0,...,I\}$, $j = \{1,...,J\}$). Furthermore, $D(i,J)$ are normalized by the length of the DTW path. After completing the calculation, the engine outputs the detection candidates, which have a normalized cost of $D(i,J)$ below a threshold θ.

3. DNN-BASED STD APPROACH

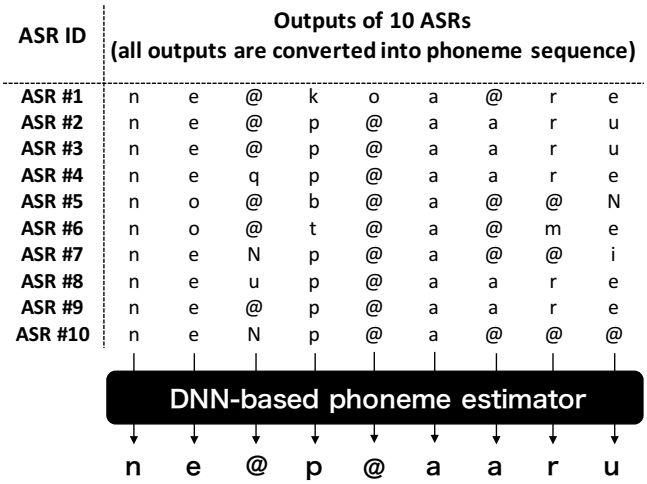

Figure 5: DNN-based correct phoneme estimation.

3.1 DNN-based phoneme estimator

Figure 5 shows the outline of the DNN-based phoneme estimator. The DNN estimator estimates the correct phoneme from phonemes belong to the same alignment. For example, it outputs the correct phoneme /u/ from the the last alignment including /e/, /u/, /N/, /i/, and deletion error, even though the majority phoneme is /e/.

Figure 6 also shows the architechture of the DNN-based phoneme estimator. The DNN-based phoneme estimator has a hidden layer, an input layer, and four hidden layers in this study. We used stacked auto-encoder for training four hidden layers. The number of nodes in each hidden layer is 1,024. We use a logistic function as an activation function in the training of output function and stochastic gradient descent (SGD) for updating parameters. The number of nodes in the output layer is 35, which is equal to the number of phoneme classes. The DNN-based phoneme estimator is trained with the 2,525 lectures in the Corpus of Spontaneous Japanese (CSJ) using Pylearn2[8].

Each phoneme-alignment has ten phonemes, including null (phoneme deletion). When a word or a phoneme is converted to a fixed dimensional vector, 1-of-N representation [17] is typically used. In this case, a phoneme can be represented as a 35-dimensional vector because we deal 35 sorts of phonemes in this study. However, we want the DNN to train phoneme-to-phoneme confusion patterns. Therefore, we convert phonemes in an alignment to a vector by considering the similarity between phonemes based on BD. The BD between phoneme p and q is calculated by the monophone-based GMMs of p and q. For the distance matrix between all the phonemes, we apply a principal component analysis to the matrix, and finally, we can get 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.

3.2 Term search engine

Figure 7 shows an example of the term search process of a query consisting of seven phonemes for a posterior-gram sequence of a target speech. The search process is very simple, just performs DP matching between a phoneme sequence of

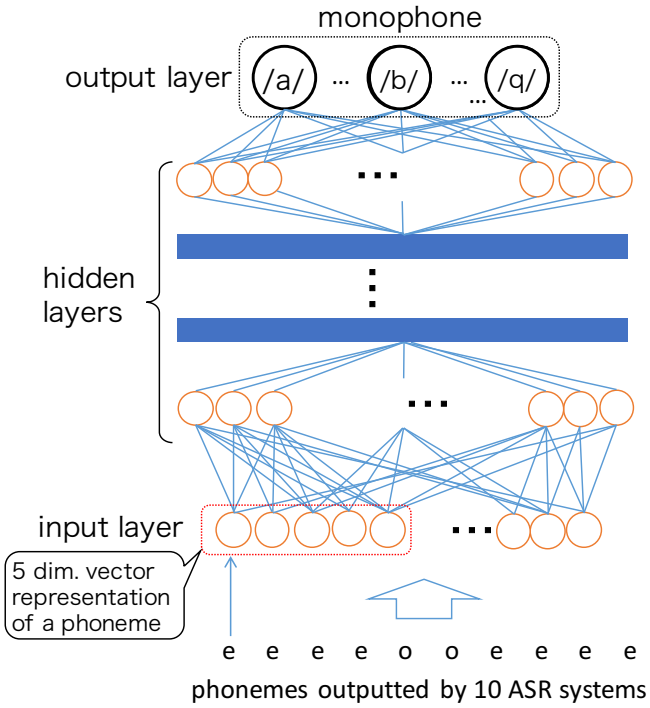

Figure 6: DNN-based correct phoneme estimation.

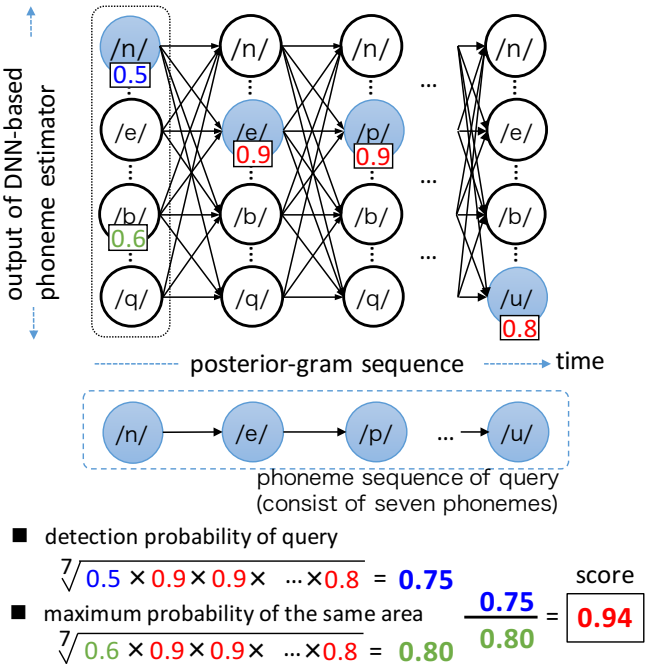

Figure 7: Term search for a posterior-gram sequence.

a query and a posterior-gram sequence.

In this example, the detection probability of the query is 0.75. At the same time, the search engine calculates the maximum probability of the query-detected region using the best probability of each posterior-gram. In this case, 0.80 is the maximum probability. The final STD score for the query is 0.94, which is obtained by dividing the detection probability by the maximum probability in the same region.

4. SYSTEM COMBINATION

We tried four sorts of score combination methods of a DTW-based score and a DNN-based STD score (same as detection probability).

One is a simple combination method as following equation, which is well-known as a weighted linear interpolation. The recomputed score $\mathrm{RS}(T, i)$ of the detection is calculated as follows:

$$\mathrm{RS}(T, i) = \gamma \cdot \mathrm{DNN}(T, i) + (1 - \gamma) \cdot \mathrm{DTW}(T, i) \quad (8)$$

where γ is a weight parameter that controls a balance between $\mathrm{DNN}(T, i)$ and $\mathrm{DTW}(T, i)$, $\mathrm{DNN}(T, i)$ and $\mathrm{DTW}(T, i)$ are scores of term T in utterance i derived by the DNN-based and the DTW-based STD methods, respectively. Both of scores from the two approaches ranges from 0 to 1. γ is determined by the moderate-size query set used in the NTCIR-10 SpokenDoc-2 [2] and the OOV test collection [9] based on the CSJ-CORE set, and common for the all query terms on the test collection.

The second combination is based on the BOSALIS toolkit [5], which is well-used for score combination on speaker recognition research field. The another method for score combination is to use Support Vecotor Machine (SVM). The SVM determines which STD system is better.

5. STD EXPERIMENT

5.1 Target test collection

The Corpus of the 1st to 7th Spoken Document Processing Workshop (SDPWS1to7) is to be used as the document collection for evaluating the NTCIR-12 SpokenQuery&Doc-2 SQ-STD subtask.

5.1.1 Speech recognition

As shown in Figure 2 and Figure 5, the SDPWS speech data is recognized by the 10 ASRs. Julius ver. 4.1.3 [10], an open source decoder for LVCSR, is used in all the systems.

We prepared two types of acoustic models (AMs) and five types of language models (LMs) for constructing the PTN. The AMs are triphone based (Tri.) and syllable based HMMs (Syl.), where both types of HMMs were trained from the spoken lectures in the Corpus of Spontaneous Japanese (CSJ) [11].

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 only by Japanese Hiragana. The words composed of 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 of Hiragana characters.

Non : No LM is used. Speech recognition without any LM is equivalent to phoneme (or syllable) recognition.

Each model is trained from the many transcriptions in the CSJ under the open for the speech data of STD.

Finally, the ten combinations, comprising two AMs and five LMs, are formed. The condition is completely the same as the description in the overview paper [4].

5.1.2 Query set of the STD subtasks

The NTCIR-12 SpokenQuery&Doc-2 organizers provided two types of query sets: the text query set and the spoken query set[4]. We evaluated our STD engine on the text query set only.

5.2 Training DNN-based model

The DNN-based phoneme estimator model was trained from a part of the CSJ except the 177 lecture speeches, which are called "CORE" [11]. A total of 2,525 lecture speeches were used to train models. We performed ASR by the 10 ASR systems described in Section 5.1.1, and all the transcriptions were translated into phoneme sequences.

5.3 STD systems

We prepared six STD systems for the NTCIR-12 STD subtask as follows:

ALPS-1 : Combination of the DTW- and DNN-based approaches by the BOSALIS toolkit and SVM.

ALPS-2 : Combination of the DTW- and DNN-based approaches by the weighted linear interpolation.

ALPS-3 : Only the DTW-based STD approach.

ALPS-4 : Combination of the DTW- and DNN-based approaches by the BOSALIS toolkit.

ALPS-5 : Combination of the DTW- and DNN-based approaches by SVM.

ALPS-6 : Only the DNN-based STD approach.

The BOSALIS toolkit and the SVM train the best combination parameter between the detections from the DTW- and the DNN-based STD systems. The training data for it is the development set from the moderate-size set of the NTCIR-10 SpokenDoc-2 STD subtask [2] and the OOV test collection [9] based on the CSJ-CORE set.

5.4 Experimental results

Table 1 and Table 2 show the summary of our STD performances. Contrary to expectation, the priority system "ALPS-1" could not get the best performance among all the submitted systems. No combination approach "ALPS-3," which is only the DTW approach, got the best performance. This is because that the combination parameter was trained with the CSJ-based test collection. In the NTCIR-12, the target speech corpus is the SDPWS lecture speeches. That is, there is the unmatched environment between the training and testing of the combination parameter. In addition, the Kaldi-based ASR systems [16] worked well for speech-recognizing the SDPWS lecture speeches. For example, word-based correct rates for the SDPWS speech was about 85%. This was adequately high ASR performance. Therefore, only the DTW-based STD approach achieved the best performance.

6. CONCLUSION

This paper introduced our STD systems and described the evaluation results on the NTCIR-12 SpokenQuery&Doc-2 SQ-STD subtask. For the NTCIR-12 evaluation, we prepared the DNN-based STD system that estimates correct phonemes from the phoneme-based transcriptions of the target speeches by the multiple ASR systems.

In the experiment on the SQ-STD test collection, our priority system could not defeat our previous proposed DTW-based system. This is why we could not optimize the combination parameter using the BOSALIS toolkit and the SVM.

In future work, we are going to explore the best parameters of the DNN-based phoneme estimator and the system combination.

7. ACKNOWLEDGMENTS

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.

8. REFERENCES

[1] M. Akbacak, L. Burget, W. Wang, and J. van Hout. Rich system combination for keyword spotting in noisy and acoustically heterogeneous audio streams. In *Proceedings of The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2013)*, pages 8267–8271, 2013.

[2] 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*, pages 573–587, 2013.

[3] 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*, pages 223–235, 2011.

[4] T. Akiba, H. Nishizaki, H. Nanjo, and G. J. F. Jones. Overview of the NTCIR-12 SpokenQuery&Doc-2 Task. In *Proceedings of the 12th NTCIR Conference*, 2016.

[5] N. Brümmer and E. de Villiers. The bosaris toolkit: Theory, algorithms and code for surviving the new dcf. In *arXiv preprint arXiv:1304.2865*, 2011.

[6] 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)*, pages 347–354, 1997.

[7] Y. Furuya, S. Natori, H. Nishizaki, and Y. Sekiguchi. Introduction of false detection control parameters in spoken term detection. In *Proceedings of the 4th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2014)*, pages 1–4, 2012.

[8] I. Goodfellow, D. Warde-Farley, P.Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio. Pylearn2: a machine learning research library. In *arXiv preprint arXiv:1308.4214*, 2013.

Table 1: Summary on the STD performances for the text query set.

	micro ave.		macro ave.				
system ID	Actual F.	Max. F.	Actual F.	Max. F.	MAP	ATWV	MTWV
ALPS-1	0.5793	0.6054	0.5965	0.6646	0.7445	0.4641	0.6420
ALPS-2	0.7099	0.7099	0.7861	0.7873	0.8401	0.7251	0.7931
ALPS-3	0.7188	0.7289	0.8258	0.8258	0.8655	0.7989	0.8242
ALPS-4	0.2780	0.5462	0.5582	0.6451	0.7224	0.3491	0.5108
ALPS-5	0.5508	0.5798	0.5645	0.6209	0.6865	0.4297	0.4962
ALPS-6	0.4399	0.4676	0.4367	0.5777	0.5164	0.3187	0.4434

Table 2: Summary on the STD performances for the spoken query set.

	micro ave.		macro ave.				
system ID	Actual F.	Max. F.	Actual F.	Max. F.	MAP	ATWV	MTWV
ALPS-1	0.4939	0.5095	0.4332	0.5841	0.6390	0.3371	0.5143
ALPS-2	0.0942	0.5606	0.0362	0.6045	0.7362	0.0291	0.5344
ALPS-3	0.0018	0.5256	0.0011	0.5261	0.7756	0.0006	0.4743
ALPS-4	0.0108	0.1828	0.0046	0.4942	0.5221	0.0037	0.1685
ALPS-5	0.4801	0.4983	0.3712	0.5232	0.5714	0.2963	0.4360
ALPS-6	0.3636	0.3872	0.3774	0.4907	0.4552	0.2646	0.3615

[9] 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 (INTERSPEECH2010)*, pages 677–680, 2010.

[10] 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 ASC2009)*, pages 131–137, 2009.

[11] K. Maekawa. Corpus of Spontaneous Japanese: Its design and evaluation. In *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR2003)*, pages 7–12, 2003.

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

[13] 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 (INTERSPEECH2008)*, pages 2146–2149, 2008.

[14] S. Natori, Y. Furuya, H. Nishizaki, and Y. Sekiguchi. Spoken Term Detection Using Phoneme Transition Network from Multiple Speech Recognizers' Outputs. *Journal of Information Processing*, 21(2):176–185, 2013.

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

[16] 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 *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU 2011)*, 2011.

[17] H. Schwenk and J.-L. Gauvai. Training Neural Network Language Models On Very Large Corpora. In *HLT/EMNLP 2005*, pages 201–208, 2005.

[18] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang. The SRI/OGI 2006 spoken term detection system. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH2007)*, pages 2393–2396, 2007.