Multiple Deep Learning Models and Architectures with Different Numbers of States Used to Improve Retrieval Accuracy of Query-by-Example

Kazuki Hatakeyama*, Masahiro Nishino†, Kazunori Kojima*, Shi-wook Lee‡, Yoshiaki Itoh*
*Iwate Prefectural University, Japan, E-mail: y-itoh@iwate-pu.ac.jp
†TOYOTA SYSTEMS CORPORATION, Japan
‡National Institute of Advanced Industrial Science and Technology, Japan, E-mail: s.lee@aist.go.jp

***Abstract*—Studies examining Spoken Term Detection (STD) and Spoken Query STD (SQ-STD) or Query by Example (QbE) using a spoken query have been conducted actively in recent years. When a spoken query is transcribed into a text using an automatic speech recognizer in SQ-STD, some misrecognition leads to retrieval accuracy deterioration. Posteriorgrams obtained using Deep Neural Network (DNN) and so on can be regarded as speaker-independent features. Although posteriorgram matching between a posteriorgram of a spoken query and posteriorgram of speech data showed high retrieval accuracy, it requires a long retrieval time and a large memory space. In earlier papers, we proposed a maximum likelihood state sequence method (MLSS) for retrieval time reduction. As described herein, we propose a method for reducing both the retrieval time and the memory space using MLSS method and multiple machine learning models with different numbers of states. The models show heterogeneous retrieval results. Their integration is probably mutually complementary and engenders retrieval accuracy improvement. Evaluation results demonstrate that the proposed method improves the retrieval accuracy, thereby reducing the retrieval time and the memory space.**

Index Terms: Query by example, Spoken term detection, maximum likelihood state sequence

I. INTRODUCTION

Studies of spoken term detection (STD), the task of finding matched sections in speech data with a query consisting of one or more words [1,2,3], have been conducted increasingly along with the increased use of storage media such as hard disk drive (HDD). NIST STD evaluation [4], Spoken Web Search (SWS) [5], QUery by Example Search on Speech Task (QUESST) [6] and NTCIR Workshop held by the National Institute of Informatics [7,8] have been applied for the evaluation of STD and Spoken Query STD (SQ-STD) [9–13]. In many STD systems, speech data to be sought are transcribed into text data using an automatic speech recognizer (ASR). Then subword sequences of the text data are compared with a subword sequence of search words (queries) at a subword level using Continuous Dynamic Programming (DP), which performs DP or DTW continuously at a frame level. Studies of Spoken Query STD (SQ-STD) using a spoken query have also been a hot topic in recent years. A spoken query is transcribed into a text using an ASR. Some misrecognition leads to deterioration of the retrieval accuracy. The representative method for SQ-STD uses posteriorgrams, sequences of posterior probability vectors for an utterance, which show speaker-independent features generated by Deep Neural Networks (DNNs) and other methods. After inputting a feature vector of a frame to DNN, posterior probabilities corresponding to the states of Hidden Markov Models (HMMs) are output for each frame. The output posterior probabilities are called posterior probability vectors. Matched sections of a spoken query among speech data are identified using continuous dynamic programming (CDP). To obtain a local distance in CDP, an inner product is computed between two posterior probability vectors of a spoken query and speech data. The inner product is transformed to a local distance by taking the negative logarithm. Posteriorgram matching generally shows high retrieval accuracy, but it requires long retrieval time and large memory space because the dimensions of a posterior probability vector amount to several thousand: about 3,000 in our experiments. For example, it took approximately 30 s and more than 100 GB to search a spoken query among approximately 30 hr of speech data. To reduce the necessary retrieval time and memory space, maximum likelihood state sequence (MLSS) for a spoken query or speech data has been proposed [14,15]. The method omits the inner product computation and enables the reduction of the retrieval time by compressing 3,000 dimensions in the posteriorgram to a single dimension. However, the retrieval accuracy of the maximum likelihood state sequence is much lower than that of posteriorgram matching because the amount of information disappears from 3,000 dimensions to a single dimension. This paper presents a balanced SQ-STD system that shows comparable performance to that of posteriorgram matching, but with greatly reduced retrieval time and memory space. The paper introduces multiple deep learning models with posterior probabilities corresponding to the states of Monophone HMMs, Triphone HMMs or Character HMMs. Each model has a different number of states or subwords (henceforth designated as the models). MLSS method is applied to these models for a spoken query and speech data. Distances obtained using the multiple models are integrated. Integration of multiple results is probably complemented mutually. It engenders the improvement of retrieval accuracy. We conduct evaluation experiments to confirm the effectiveness of the introduction of multiple models using open test collections.

II. Related Work

A. QbE using Posteriorgram

In QbE, posteriorgram matching is a representative method that searches a posteriorgram of a spoken query for posteriorgrams of speech data using Dynamic Time Warping (DTW) or Continuous Dynamic Programming (CDP) that performs DTW continuously at a frame level [16]. A posterior probability vector is generated by Deep Neural Network (DNN) at each frame and each posterior probability corresponds to a likelihood of a state of triphone acoustic models. An inner product of the two posterior vectors is calculated to obtain a measure of similarity between two posterior vectors. Therefore, the negative logarithm of the inner product transforms the similarity to a local distance. A local distance is therefore calculated as shown below.

$$\mathrm{D}\left(P_i^d, P_j^q\right) = -log_{10}\left(\sum_{k=0}^{N} P_i^d\,(k)\ \cdot P_j^q(k)\right) \qquad (1)$$

Therein, P_i^d , P_j^q, N, and k respectively denote a posterior probability vector of the i-th frame in speech data, a posterior probability vector of the j-th frame in a speech query, the number of dimensions of the posterior probability vector, and the k-th dimension in a posterior probability vector.

Subword matching method, another representative approach for QbE, converts speech data to subword sequences using ASR beforehand. A spoken query is also converted to a subword sequence. A subword sequence of the spoken query is sought among subword sequences of speech data. Comparison of posteriorgram matching with the subword matching method reveals that, although posteriorgram matching shows high search accuracy, it requires much more retrieval time to calculate the inner product of about 3,000 dimensions. It also requires huge memory size: number of frames of speech query × number of frames of speech data.

B. Spoken query / Speech data maximum likelihood state sequence method

A spoken query / speech data maximum likelihood state sequence (MLSS) [14,15] method was proposed to reduce the huge memory size and long retrieval time of posteriorgram matching. The MLSS method is explained briefly because the proposed method uses MLSS for a spoken query and speech data. An image of MLSS method for speech data is presented in Figure 1. At each frame in posteriorgrams of the speech data, the state number that shows the maximum probability in a posterior probability vector is extracted. A posterior probability vector with around 3,000 dimensions is compressed to a single dimension of which a sequence is a so-called a maximum likelihood state sequence (MLSS). The MLSS (Fig. 1 upper right) is held in advance. Actually, MLSS shows the sequence of the most probable state number for the speech data. Given a spoken query, it is converted to posteriorgram as speech data, as shown in the lower left panel in Figure 1. A posteriogram of a spoken query is located on the vertical axis. MLSS is placed at the horizontal axis in the lower right panel. At each frame of speech data, posterior probabilities corresponding to a maximum state number are referred to posterior probabilities of the same state number in the posteriorgram of the spoken query. For example, if the state number is 1 in the right orange rectangle in speech data, then the probabilities at state 1 in the posteriorgram of a spoken query are referred (left orange rectangle). Results indicate that when a sequence similar to the spoken query in speech data is found, high posterior probabilities are found diagonally between the corresponding frames, as shown in the lower right panel. Each posterior probability can be transformed into a distance beforehand according to equation (2). Therefore, calculation of the local distances is not required in the process of DTW or CDP. The posterior probability matrix is replaced by a local distance matrix. Letting the state of the frame i of the speech data be $s(i)$, the local distance of the point (i, j) that denotes the frame j in spoken query is obtained by referring to $(s(i),j)$ in the local distance matrix of the spoken query.

$$D(s_{(i)}, \mathrm{j}) = -log_{10}(P_{(i,j)}) \qquad (2)$$

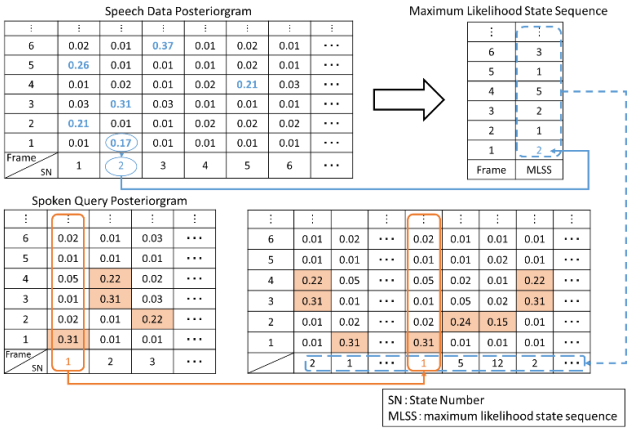

Fig. 1. Speech data maximum likelihood sequence method flow

The MLSS for a spoken query constructs a local distance matrix after a spoken query is given; computation of local distances by equation (2) is small.

Because posteriorgrams of the speech data with about 3,000 dimensions are converted to the maximum likelihood sequence of a single dimension, the memory size was reduced by about 99%. The retrieval time was around 1/10 compared to that of posteriorgram matching, but the retrieval accuracy is lower because of the decreased amount of information.

III. Proposed Method

As described in this section, we apply maximum likelihood serialization method to multiple deep learning models and integrate the obtained retrieval results to achieve high accuracy, high speed, and low memory retrieval compared with posteriorgram matching. Each model and the reason for its use are explained below.

A. BLSTM

A network of bi-directionally connected LSTMs, BLSTM, is an extension of the Recurrent Neural Network (RNN) structure. Compared to unidirectional learning models such as DNNs, BLSTM can learn bidirectional features. It is expected to improve the retrieval accuracy. BLSTM is used for this study, where the posterior probability output by BLSTM corresponds to each state of a triphone.

B. ESPnet

In recent years, end-to-end learning models have been used for ASRs. Reportedly, they provide higher recognition accuracy than DNN-HMM hybrids and other speech recognition systems [17]. The end-to-end ASR can map speech features to words and subwords directly through training. Moreover, it requires no correct label for each frame, which is necessary for conventional hybrid ASRs. Because ESPnet can construct models flexibly with different states corresponding to speech features such as words and subwords, ESPnet is used for the proposed method to construct models with heterogeneous state numbers. We apply maximum likelihood state sequence method (MLSS) to the models. ESPnet composed of Hybrid CTC/Attention [18] was used as an end-to-end speech recognizer. The same encoder is used for both CTC and Attention as a shared encoder. As described herein, we extract posteriorgrams, which comprise posterior probability vector output from CTC for input speech features. The heterogeneous models correspond to posterior probabilities that denote characters, syllables, and monophones.

C. Proposed method using heterogeneous and multiple models

When multiple deep learning models are used for QbE, each model generates a score for each utterance. The score denotes the distance described in Chapter 2 in this paper. A single model among the multiple models might give incorrect scores. For multiple models, one can assume that one model using heterogeneous models outputs the correct score, and that the score can be optimized using multiple scores [19].

Although, we concatenated posteriorgrams of each model such as BLSTM and CTC in our previous experiments, the retrieval accuracy was not improved. As described herein, we use heterogeneous models with posterior probabilities corresponding to characters, syllables, triphones, and monophones so that multiple scores are obtainable from the heterogeneous model for each utterance. The practical search time and memory size are investigated during QbE. The proposed system is aimed at a balanced system among retrieval accuracy, search time, and memory size for QbE.

D. Score integration

Scores obtained from the retrieval of heterogeneous and multiple models were integrated linearly. Given a speech query, two scores denoted as D_1 and D_2 were obtained from the two models for an utterance. The new score D_{new} was obtained using Equation (3). Weighting factor α was set to $0 \leq \alpha \leq 1$ and was determined using a cross-validation method with two test sets.

$$D_{new} = \alpha D_1 + (1 - \alpha) D_2 \quad (3)$$

IV. Evaluation Experiments

A. SQ-STD using Posteriorgram

BLSTM and Hybrid CTC/Attention were trained using the Corpus of Spontaneous Japanese (CSJ), which contains 2,702 lectures: about 600 hr speech. Features for BLSTM are 120 dimensions composed of 40 dimensional filter bank (FBANK) and its Δ, and ΔΔ. The dimension of the input feature amounts to 1,320 of 11 frames, adding 5 frames before and after the current frame. BLSTM output layers correspond to each state of triphone. Triphones share the state and the number of states amounts to 3,009. Features for Hybrid CTC/Attention architecture are 83 dimensions, which are 80 dimensions of filter bank (FBANK) adding three dimensions of pitch. For Hybrid CTC/Attention, we prepare three models with output layers corresponding to characters, syllables, and monophones. The numbers of output nodes for the three models were, respectively, 3,245, 264, and 43 (3,242 words, 261 syllables, 40 phonemes adding three types of labels: blank label, <unk> label for unknown state, and <eos> label for terminal state appearing in the training data). The processing time was measured using a personal computer: CPU, Core i7-6700K, Intel Corp.; GPU, GeForce GTX 1080, NVIDIA; memory, 16 GB. The retrieval using multiple models are conducted in parallel on multiple CPUs.

TABLE I
Conditions for Feature Extraction

	DNN, BLSTM, CTC	Hybrid CTC / Attention
Feature parameter	120 dimensions FBANK (40 dims) + ΔFBANK (40 dims) + ΔΔFBANK (40 dims)	83 dimensions (FBANK (80 dims)) + Pitch (3 dims)
Window length	25 ms	25 ms
Frame shift	10 ms	10 ms

TABLE II
Two Open Test Collections

	NTCIR-10	NTCIR-12
Spoken documents	104 presentations, 29 hr, 40,746 utterances	98 presentations, 27.5 hr, 37,782 utterances
Query sets	Formal run: 100 (10 people)	Formal run: 113 (10 people)

B. Test Collections

The NTCIR-10 Formal run and NTCIR-12 Formal run shown in Table II were used as test sets for experimental evaluation. NTCIR-10 includes 104 lectures (about 29 hr, 40,746 utterances) of the Spoken Document Processing Workshop (SDPWS). NTCIR-12 includes 98 lectures (about 29 hr, 37,782 utterances) of SDPWS. NTCIR-10 and NTCIR-12 respectively include 100 and 113 queries and the correct labels of the queries. Because NTCIR-10 includes no spoken query, we

recorded 100 queries uttered by 10 people (5 men and 5 women); all 1000 utterances are used for spoken queries. Spoken queries provided by the organizer were used for NTCIR-12. Mean average precision (MAP) was used to evaluate the retrieval accuracy.

C. Retrieval Accuracy of Each Single Model

Table III presents the retrieval performance (retrieval accuracy, retrieval time, and memory size) of each trained single model using posteriorgram matching and MLSS methods. The retrieval accuracy of posteriorgram matching of BLSTM was 79.78% for NTCIR-10 and 74.83% for NTCIR-12, which is used as the baseline in this paper. Because the Hybrid CTC/Attention using monophone has small dimensionality (43 dimensions) and because it requires no large amount of memory size, MLSS for a spoken query was used. For BLSTM, half of the frames are used to reduce the retrieval time by averaging the posterior probability values of two neighboring frames. Although the retrieval accuracy was reduced by an average of 3 pts, the retrieval time and the required amount of memory were reduced by half.

The highest retrieval accuracy obtained using MLSS method was for NTCIR-10 with Hybrid CTC/Attention of syllable, using MLSS for speech data, and for NTCIR-12 with Hybrid CTC/Attention of monophone using MLSS for a spoken query.

TABLE III
Retrieval Accuracies for the Respective Models and Architectures

NTCIR-10					
Retrieval method	Posteriorgram	Maximum likelihood			
		MLSS for Speech data			MLSS for Spoken query
Model	BLSTM	BLSTM	Hybrid CTC/Attention		
			Character	Syllable	Monophone
Retrieval accuracy (%)	79.78	69.92	71.86	78.00	77.96
Retrieval time (s)	29.08	1.84	0.23	0.20	0.37
Memory capacity (GB)	114	0.01	0.006	0.006	0.41

NTCIR-12					
Retrieval method	Posteriorgram	Maximum likelihood			
		MLSS for Speech data			MLSS for Spoken query
Model	BLSTM	BLSTM	Hybrid CTC/Attention		
			Character	Syllable	Monophone
Retrieval accuracy (%)	74.83	66.48	72.37	75.74	79.51
Retrieval time (s)	27.35	1.69	0.20	0.20	0.31
Memory capacity (GB)	107	0.01	0.005	0.005	0.38

Table IV presents comparisons of the retrieval accuracy of the proposed method using four models (all) and posteriorgram matching. These four models are shown on the right side of Table III (BLSTM–character–syllable–monophone). The integration ratio was changed by 0.1. The best retrieval accuracies were obtained with (0.2–0.1–0.2–0.5) and (0.1–0.4–0.3–0.2), respectively, for NTCIR-10 and NTCIR-12. The other best integration ratio was used in a cross-validation case. For example, the best integration ratio of NTCIR-12 was used for the experiment of NTCIR-10 in a cross-validation case. The retrieval accuracy of the proposed method using four models was 85.17% with +5.39 pt in NTCIR-10 compared with 79.78% in BLSTM posteriorgram matching, and 84.94% with +5.16 pt in cross-validation in a cross validation case, which exceeded the accuracy of posteriorgram matching. The retrieval accuracy of the proposed method using four models was 85.25% with +10.42 pt in NTCIR-12, compared to 74.83% in BLSTM posteriorgram matching and 85.19% with +10.36 pt in cross-validation in a cross validation case, which is much better than the accuracy achieved with posteriorgram matching. Cross-validation led to no decrease in the retrieval accuracy for either dataset. The integration ratio therefore, didn't affect the retrieval accuracy seriously.

The search time was calculated by actually running all models in parallel on one PC and by adding the integration time (0.08 s for NTCIR-10 and 0.07 s for NTCIR-12). Compared with the retrieval time of posteriorgram matching of BLSTM, 29.08 s was reduced to 2.11 s for NTCIR-10. The same tendency was observed for NTCIR-12. The required memory size during the retrieval was reduced greatly to about 0.4 GB for both NTCIR-10 and NTCIR-12.

TABLE IV
Comparison of integrated results with those obtained by posteriorgram matching

	NTCIR-10		NTCIR-12	
Retrieval method	Posteriorgram	Integration	Posteriorgram	Integration
Model	BLSTM	ALL	BLSTM	ALL
Retrieval accuracy (%)	79.78	**85.17**	74.83	**85.25**
Retrieval time (s)	29.08	**2.11**	27.35	**1.91**
Memory capacity (GB)	114	**0.43**	107	**0.40**

V. CONCLUSIONS

As described in this paper, we integrate the scores obtained from four deep learning models linearly with different corresponding states in the SQ-STD task to improve the retrieval accuracy while considering the practical retrieval time and the memory size. Results of integration demonstrated the retrieval time as about 2 s, even with the four models integrated. The necessary memory size for retrieval was about 0.4GB for both datasets. The retrieval accuracy was 84.94% in the cross-validation case (the best percentage was 85.17%) for NTCIR-10, and 85.19% in the cross-validation case (the best percentage was 85.25%) for NTCIR-12. In both datasets, the retrieval accuracy was more than 5 percentage points higher than that of posteriorgram matching of BLSTM. Cross-validation led to no decrease in the retrieval accuracy for either dataset.

ACKNOWLEDGMENT

This work was supported by JSPS (C), KAKENHI Grant Number 21K12611.

REFERENCES

[1] C. Auzanne, JS. Garofolo, JG. Fiscus, and WM Fisher, "Automatic Language Model Adaptation for Spoken Document Retrieval," 2000TREC-9 SDR Track, 2000.

[2] A. Fujii, and K. itou, "Evaluating Speech-Driven IR in the NTCIR-3Web Retrieval Task," Third NTCIR Workshop, 2003.

[3] P. Motlicek, F. Valente, and PN. Garner, "English Spoken Term Detection in Multilingual Recordings," INTERSPEECH, pp.206-209, 2010.

[4] OpenKWS13 Keyword Search Evaluation Plan, 2013 http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-evalplan-v4.pdf

[5] The 2013 Spoken Web Search Task, MediaEval Benchmarking Initiative for Multimedia Evaluation, 2013, http://www.multimediaeval.org/mediaeval2013 /sws2013.

[6] Igor Szöke, Luis Javier, Rodriguez-Fuentes, Andi Buzo, Florian Metze, Xavier Anguera, "Query by Example Search on Speech Task (QUESST 2015)," http://www.slideshare.net/multimediaeval/mediaeval-2015-query-by-example-search-on-speech-task

[7] Tomoyosi Akiba et al, "Overview of the NTCIR-10 SpokenDoc Task," Proceedings of the 10th NTCIR Conference, pp.573-587, 2013.

[8] Tomoyosi Akiba et al, "Overview of the NTCIR-11 SpokenQuery&Doc Task," Proceedings of the 11th NTCIR Conference, pp.350-364, December 2014.

[9] Jonathan G. Fiscus et al, SIGIR Workshop Searching Spontaneous Conversational Speech. Results of the 2006 spoken term detection evaluation, pp. 45-50, 2007.

[10] Tomoyosi Akiba et al, "Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop," NTCIR-9 Workshop Meeting, pp. 223-235, 2011.

[11] Tomoyosi Akiba et al., "Overview of the NTCIR-10 SpokenDoc-2 Task," NTCIR-10 Workshop Meeting, pp. 573-587, 2013.

[12] Tomoyosi Akiba et al, "Overview of NTCIR-11 Spoken&Doc Task, NTCIR-11," pp. 350-364, 2014.

[13] Tomoyoshi Akiba et al, "Overview of NTCIR-12 Spoken&Doc-2 Task," NTCIR-12 Workshop Meeting, pp.167-179, 2016.

[14] Daisuke Kaneko, Ryota Konno, Kazunori Kojima, Kazuyo Tanaka , Shi-wook Lee and Yoshiaki Itoh, "Constructing Acoustic Distances between Subwords and States Obtained from a Deep Neural Network for Spoken Term Detection," INTERSPEECH, pp.2879-2883, 2017.

[15] Takashi Yokota , Kazunori Kojima , Shi-wook Lee and Yoshiaki Itoh, "Reduction of Speech Data Posteriorgrams by Compressing Maximum-likelihood State Sequences in Query by Example," Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2020.

[16] Yoshiaki Itoh and Kazuyo Tanaka, "Frequent word section extraction in a presentation speech by an effective dynamic programming algorithm," The Journal of the Acoustical Society of America, vol.116 No.2, pp. 1234-1243(2004-8).

[17] Shinji Watanabe et al, "ESPnet: End-to-End Speech Processing Toolkit," INTERSPEECH, pp.2207-2211, 2018.

[18] Shinji Watanabe et al, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240–1253, 2017.

[19] Shi-wook Lee, Kazuyo Tanaka, Yoshiaki Itoh, "Effective Combination of Heterogeneous Subword-based Spoken Term Detection Systems," 4 pages, IEEE Spoken Language Technology Workshop (SLT), 2014-12.