Introducing Self-Supervised Learning Models for Spoken Query-Spoken Term Detection

Masato Nagase * Kazunori Kojima* Shi-wook Lee‡ Yosiaki Itoh*
* Iwate Prefecture University, Japan E-mail: y-itoh@iwate-pu.ac.jp
‡ National Institute of Advanced Industrial Science and Technology, Japan, E-mail: s.lee@aist.go.jp

***Abstract*—This study was undertaken to improve the performance of Spoken Query Spoken Term Detection (SQ-STD) by introducing Self-Supervised Learning (SSL) models. To construct posteriorgrams, the SSL models are augmented with a Connectionist Temporal Classification (CTC) layer and extracted posterior probability vectors, which represent phonemes or syllables from the layer immediately preceding the CTC output layer. Posteriorgram matching is performed using posteriorgrams generated from SSL models of three types known for their high speech recognition accuracy: wav2vec 2.0, HuBERT, and WavLM. Experiment results obtained on the NTCIR evaluation sets demonstrate that our approach achieves retrieval accuracy exceeding 92%, which is the state of the art for the evolution sets.**

I. Introduction

In recent years, Spoken Query Spoken Term Detection (SQ-STD) has gained wide attention as a method for locating specific scenes within large-scale multimedia data including speech. The SQ-STD task aims at detecting sections in target speech data where a spoken query term, also given in audio form, is uttered.

A very well-known approach applied for SQ-STD is Posteriorgram matching [1]. Using this method, after acoustic features are extracted frame-by-frame from both the spoken query and the target audio data, they are input into a deep learning model to obtain posterior probability vectors of speaker-independent acoustic units such as phonemes. By arranging these vectors in temporal order, a matrix known as a posteriorgram is trained and constructed. Local distances are found by computing the inner product between the posterior vectors of the query and the target audio in Posteriorgram matching, followed by continuous dynamic programming (DP) that applies Dynamic Time Warping with Constraints (DTWC) to perform alignment and to detect matching segments.

In an earlier study, a method using system ESPnet [2], which deployed syllables, achieved state-of-the-art retrieval accuracy, processing speed, and memory efficiency for subwords with Posteriorgram matching. Also, ESPnet employs a Hybrid CTC/Attention architecture, where features from the encoder are passed to both a Connectionist Temporal Classification (CTC) [3] decoder and an attention-based decoder. The posterior features obtained from these decoders are used to predict character sequences. They are believed to include meaningful acoustic representations such as phonemes and syllables. In earlier work, Posteriorgram matching using CTC-derived features achieved retrieval accuracy that exceeded 80%.

For this study, we introduce self-supervised fine-tuned (SSL) models, which are used widely for recent speech recognition research, into the Posteriorgram matching framework. Recent SSL models demonstrate higher character recognition accuracy than those of existing models such as ESPnet. The SSL models used for this study are wav2vec 2.0 [4], HuBERT [5], and WavLM [6]. By fine-tuning SSL models for subword units and by applying them to Posteriorgram matching, we aim to improve the SQ-STD accuracy. We hypothesize that using these SSL models for Posteriorgram matching achieves higher retrieval accuracy than Posteriorgram matching based on ESPnet. Unlike earlier SQ-STD studies, which rely primarily on end-to-end ASR architectures (e.g., ESPnet) or unsupervised unit discovery [2], our approach explicitly integrates **fine-tuned self-supervised encoders with a CTC-based subword decoder** to generate posterior features tailored for Posteriorgram matching. Although earlier methods used either character-level decoders or relied on built-in subword modeling of ASR systems, our framework decouples the encoder and decoder by appending a lightweight CTC module trained specifically for **phoneme or syllable recognition [7–9]**, enabling more fine-grained control over subword representation. This separation enables us to leverage the powerful representations learned by SSL models while customizing the output space for effective STD performance. Although we attempted Posteriorgram matching using text-based SSL models in our preliminary experiments, these models were inadequate to exceed the retrieval accuracy achieved by ESPnet. Therefore, we adopt an approach by which a CTC module is appended downstream of the SSL encoder and is fine-tuned for subwords such as monophones and syllables. The posterior probability vectors obtained from the output layer of the trained CTC module are then used for Posteriorgram matching. These vectors, when arranged as a Posteriorgram, are expected to capture acoustic information effectively.

As described herein, we achieve both high accuracy and low memory usage by application of the Blank-cut method [10] proposed in earlier studies. The method, which removes blank frames in the Posteriorgram created by SSL and the frame duplication removal method [11], compresses consecutive

979-8-3315-7206-8/25/$31.00

blank and same-phoneme frames into a single frame.

Experiment results obtained using the NTCIR evaluation dataset, which is a Japanese open dataset, demonstrate that our approach outperforms the state-of-the-art ESPnet-based system, which achieved superior retrieval accuracy (80.19%). All three SSL-based Posteriorgram matching methods exceeded this baseline, with HuBERT achieving retrieval accuracy higher than 91%. The proposed method using SSL models improved the retrieval performance compared to conventional systems.

II. Earlier Research

First, as a representative SQ-STD method, we describe the Posteriorgram matching method. Then, the Blank-cut and frame duplication removal techniques are described, which were proposed to reduce memory usage and to achieve high retrieval accuracy.

A. *Posteriorgram Matching*

Vectors of posterior probabilities such as those of phonemes are obtained by inputting acoustic features into a deep learning model. **A posteriorgram** is a matrix formed by arranging those vectors in frame order. By **Posteriorgram matching**, as shown in Figure 1, the posteriorgram of a spoken query is compared to that of the speech data using continuous dynamic programming (DP), which applies DTW continuously, or similar methods to detect the segment in which the query is spoken. The local distance during matching is calculated by taking the inner product of the posterior probability vectors from both the spoken query and the speech data to obtain a similarity measure, and by then taking the negative logarithm of this value.

B. *Blank Cut*

In the CTC framework, **blank tokens,** which do not correspond to any actual character are inserted primarily to align the length of the input (i.e., number of acoustic frames) with that of the output (i.e., number of characters) and to

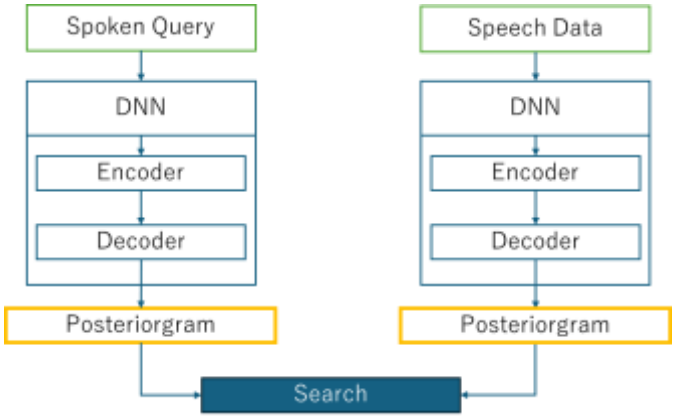

Fig. 1 SQ-STD using posteriorgram matching.

distinguish between repeated characters during decoding. Blank tokens contribute minimally to the Posteriorgram matching effectiveness, but require approximately **90% of memory usage**. By contrast, **Blank cut is proposed**, which removes blank tokens from the posteriorgram during the matching process to reduce memory usage and to accelerate computation. Although applying Blank cut leads to slight degradation in retrieval accuracy (by a few percentage points), it is expected to reduce memory consumption considerably, **by up to one-seventh**.

C. *Frame De-Duplication*

In the CTC framework adopted for this study, the same **tokens often appear consecutively** in the output sequence. Such repeated segments are generally interpreted by the decoder as **prolonged pronunciations of the same speech.** They are treated as a single phonetic unit during decoding. Therefore, it is assumed that **collapsing** these redundant frames in the posteriorgram during matching has only minimal effects on retrieval performance. Based on this assumption, **Frame De-duplication (FDD)** was proposed, which removes consecutive duplicate frames from the posteriorgram to reduce memory consumption and to accelerate computation. By application of this method, it is expected that **memory usage can be reduced by up to one-fifth** while maintaining retrieval accuracy comparable to that achieved without FDD.

III. Proposed Method

A. *Introduction SSL for Posteriorgram Construction*

As described in this paper, we propose a method that introduces self-supervised learning (SSL) models used for constructing posteriorgrams, with the aim of improving SQ-STD accuracy. SSL consists of a two-stage training process: a **pre-training** phase using a large amount of unlabeled data, during which pseudo-labels are generated automatically, followed by **fine-tuning** using a small set of labeled data. For the work presented herein, we introduce three state-of-the-art SSL models: **wav2vec 2**, **HuBERT**, and **WavLM**. These models have demonstrated higher character recognition accuracy than conventional models. Then we hypothesize that introduction of those models for posteriorgram construction can yield higher retrieval accuracy.

B. *CTC Training for Phoneme-level Representations*

Preliminary experiments revealed that high retrieval accuracy was not obtained when intermediate features from the encoders of these pre-trained SSL models are used as posteriorgrams. Additionally, we experimented with Posteriorgram matching using character-based pre-trained models, but these also were inadequate to surpass the performance of existing Posteriorgram matching methods.

Therefore, we appended a Connectionist Temporal Classification (CTC) module to the downstream of each SSL model. As presented in Figure 2, this module is designed to output Japanese phonemes, syllables, or similar linguistic units. After we fine-tuned the entire system using the Corpus of Spontaneous Japanese (CSJ), which includes approximately 600 hours of speech data, we used the posterior probability vectors obtained from the CTC as the Posteriorgram representation.

IV. Evaluation Experiments

Fig. 2 proposed method: posteriorgram matching using a model with a CTC layer attached to the output of SSL encoder

A. *Experiment Conditions*

The experiment compares the results of Posteriorgram matching using the proposed method employing wav2vec 2, HuBERT, and WavLM with those obtained using earlier reported methods that achieved the highest retrieval accuracy using BLSTM and ESPnet on the NTCIR evaluation set. Because recent state-of-the-art methods such as SSL-based approaches have not been evaluated on the NTCIR-10 and NTCIR-12 datasets, we use the best-performing ESPnet-based system as the baseline for comparison. The training conditions for each model are presented in Table 1. For each of the monophone and syllable settings, the model which achieved the highest accuracy (i.e., the lowest WER on inference data) within 30 training epochs was selected. Among these, the one with the higher MAP score was used for evaluation.

For evaluation, the Japanese open test sets NTCIR-10 and NTCIR-12 are used. Also, Mean Average Precision (**MAP**) is used as the evaluation metric for retrieval accuracy. MAP is the average of Average Precision (**AP**). AP is defined as the average of the **precision** values at the ranks where relevant (i.e., correct) results are returned for a given query.

Table 1 Conditions for each model

		BLSTM	ESPnet	wav2vec2	HuBERT	WavLM
Training Data	Pre-Training	CSJ (600h)	CSJ (600h)	ReasonSpeech (17,000 h)	ReasonSpeech (17,000 h)	LibriLight (60,000 h)
	Fine-Turning			CSJ (600 h)	CSJ (600 h)	CSJ (600 h)
Input		FBANK	FBANK	Waveform	Waveform	Waveform
Output		Character	syllable	monophone	syllable	syllable
dim		3009	264	42	263	263
epoch		30	26	3	10	30

B. *Comparison among SSL, BLSTM, and ESPnet Results*

First, the results of Posteriorgram matching using NTCIR-10 are presented in Figure 3. The bar graph represents MAP.

Table 2 Test dataset

Evaluation Set	NTCIR-10	NTCIR-12
Search corpus	SDPWS104 lec	SDPWS98 lec
Number of Queries	100	113
Duration (s)	361.6	354.4
Number of Documents	104	98

The line graph in the figure shows memory usage.

In the proposed method introducing SSL, higher accuracy was achieved compared to either ESPnet or BLSTM. Among the models tested, HuBERT yielded the highest accuracy. All three SSL models outperformed those of earlier studies in terms of individual performance. Next, the results of Posteriorgram matching obtained when using NTCIR-12 are presented in Figure 4, which observed in NTCIR-10.

Fig. 3 Matching results on NTCIR-10

Fig. 4 Matching results on NTCIR-12.

Because memory usage is proportional to the number of frames and output dimensions in the posteriorgram, wav2vec2, which has fewer output dimensions, required the least memory consumption. Although the numbers of frames were

Table 3 Results of all matching experiments conducted in this study on the NTCIR-10 and NTCIR-12 datasets, including baseline methods (BLSTM, ESPnet) and the proposed methods (wav2vec2, HuBERT, WavLM)

		Posteriorgram					Posterorgram+Blank-cut				Posterorgram+FDD			
model		BLSTM	ESPnet	wav2vec2	HuBERT	WavLM	ESPnet	wav2vec2	HuBERT	WavLM	ESPnet	wav2vec2	HuBERT	WavLM
NTCIR-10	MAP(%)	79.28	75.31	86.70	**92.38**	91.39	83.78	73.94	**89.36**	88.62	80.37	77.87	**91.94**	91.66
	Memory(GB)	114.64	2.47	**0.80**	5.02	5.02	1.16	**0.22**	0.82	0.82	1.02	**0.33**	1.77	1.77
NTCIR-12	MAP(%)	74.83	80.19	84.62	**91.60**	91.02	81.84	72.36	**86.00**	84.85	79.23	75.87	89.76	**90.41**
	Memory(GB)	107.34	2.31	0.75	4.70	4.70	1.04	**0.21**	0.76	0.77	1.29	**0.31**	1.45	1.46

nearly equal across SSL models, the output dimensionality of HuBERT and WavLM was about six times that of wav2vec2, leading to approximately six times greater memory usage.

C. Comparison of Results with Blank Cut Applied

Next, we present matching results obtained when applying the Blank-cut method to the posteriorgram. This experiment was conducted using the ESPnet and SSL models, both of which produce blank tokens in their posteriorgrams.

Figure 5 presents results of Posteriorgram matching using the NTCIR-10 dataset. The proposed method introducing (SSL) models achieved higher accuracy than ESPnet. Among the three models, HuBERT showed the highest accuracy. All SSL models outperformed earlier studies in individual model performance. However, unlike ESPnet, applying Blank cut to SSL led to lower retrieval accuracy. Figure 6 presents Posteriorgram matching results obtained when using the NTCIR-12 dataset. Similar trends were observed to those obtained for NTCIR-10.

Fig. 5 Matching results with the Blank-cut method on NTCIR-10.

Fig. 6 Matching results with the Blank-cut method on NTCIR-12.

Unlike results obtained from Posteriorgram matching, memory usage in this experiment was lower for HuBERT and WavLM than for ESPnet. From these results, it can be inferred that HuBERT and WavLM have a higher proportion of blanks in their posteriorgrams than ESPnet has. This difference in the proportion of blank frames is believed to have contributed to the decline in retrieval accuracy obtained when applying Blank cut to SSL-based matching.

D. Comparison of Results with FDD Applied

This experiment was conducted using the ESPnet and SSL models, both of which include blanks in their posteriorgrams.

Figure 7 presents results obtained using NTCIR-10. The proposed method introducing SSL achieved higher accuracy than ESPnet. Among the three SSL models, HuBERT showed the highest accuracy. All three models individually outperformed results obtained from earlier studies. Similarly to Blank cut, the application of FDD also led to decreased

Fig. 7 Matching results with the FDD method on NTCIR-10.

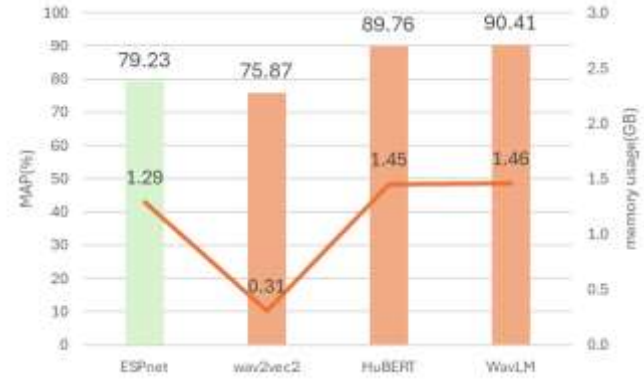

Fig. 8 Matching results with the FDD method on NTCIR-12.

retrieval accuracy for SSL models, in contrast to ESPnet. Figure 8 presents the matching results obtained using NTCIR-12. Results obtained for NTCIR-12 demonstrated that WavLM achieved the highest accuracy.

Additionally, the memory usage of HuBERT and WavLM was higher than that of ESPnet.

V. CONCLUSION

As described herein, for Posteriorgram matching applied for spoken term detection, we introduced SSL models of three types: wav2vec2, HuBERT, and WavLM. By connecting a CTC layer after the encoder and by training on subword sequences, we strove to improve accuracy. Experiments using the NTCIR evaluation set showed that posteriorgram matching with HuBERT achieved search accuracy exceeding 91%, thereby surpassing the best result obtained earlier by ESPnet: 80.19%. The memory usage of wav2vec2 was comparable to that of ESPnet, whereas HuBERT and WavLM required more memory. Additionally, when applying Blank-cut and FDD techniques for matching, the search accuracy of the SSL models was lower than with standard matching, but it still outperformed ESPnet. As shown in Table 3, these results demonstrated that applying Blank cut to the posteriorgram generated by HuBERT improves both retrieval accuracy and memory efficiency compared to ESPnet. This finding confirms that Posteriorgram matching using SSL models is an effective approach for SQ-STD.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI Grant Number 25K15177

REFERENCES

[1] [R. Konno, et al., “Application of subword-level acoustic distance based on DNN posterior probabilities to spoken term detection,” in *Proc. Spring Meeting of the Acoustical Society of Japan*, 2015.

[2] H. Lu, et al., “ESPnet: End-to-End Speech Processing Toolkit,” *Proc. Interspeech*, 2019.

[3] 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.

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” *NeurIPS*, 2020.

[5] W.-N. Hsu, et al., “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units,” *IEEE/ACM Trans. ASLP*, 2021.

[6] S. Chen, et al., “WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing,” *IEEE J. STSP*, 2022.

[7] P. M. Reuter, C. Rollwage, and B. T. Meyer, “Multilingual Query-by-Example Keyword Spotting with Metric Learning and Phoneme-to-Embedding Mapping,” in IEEE ICASSP, 2023, pp. 1–5.

[8] L. Lugosch, S. Myer, and V. S. Tomar, “DONUT: CTC-based Query-by-Example Keyword Spotting,” arXiv preprint arXiv:1811.10736, 2018.

[9] Z. Xiao, Z. Ou, W. Chu, and H. Lin, “Hybrid CTC-Attention based End-to-End Speech Recognition using Subword Units,” in Proc. ISCSLP, 2018.

[10] M.Nishino, et al., “High-accuracy, fast, and memory-efficient spoken term detection using multiple deep learning models,” master’s thesis, Tokyo Institute of Technology, 2020. (in Japanese)

[11] Yusuke Hatakeyama, Kazunori Kojima, Shixu Li, and Yoshiaki Ito,“Improving Search Accuracy, Speed, and Memory Usage for Spoken Term Detection Using Multiple Deep Learning Models and Frame Compression Techniques,”IPSJ Transactions on Information and Systems, 2023.