Query-by-Example Spoken Term Detection using Attentive Pooling Networks

Kun Zhang*, Zhiyong Wu*, Jia Jia†, Helen Meng‡, Binheng Song*
* Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
E-mail: zk17@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn, songbinheng@sz.tsinghua.edu.cn
† Beijing National Research Centre for Information Science and Technology (BNRist),
Department of Computer Science and Technology, Tsinghua University, Beijing, China
E-mail: jjia@tsinghua.edu.cn
‡ Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China
E-mail: hmmeng@se.cuhk.edu.hk

Abstract—**Query-by-example spoken term detection (QbE-STD) is attractive because its a key technology for retrieving and browsing spoken content without transcribing them into text. Several end-to-end models based on encoder architecture have been proposed for QbE-STD, in which the input pair, spoken query and audio segment, are first projected into fixed-length vector representations by feature extraction module and then similarity measure module is used to output detection score based on the representations. Attention mechanism has been applied into the feature extractor; however, traditional approach calculates attention vector for audio segment only, which makes it a one-way attention mechanism. In this paper, we present a novel feature extraction module based on two-way attention mechanism, called attentive pooling networks, for end-to-end QbE-STD. The main idea is to learn a similarity measure over the projected input pair and extract information in a way that two input items can directly influence the computation of each other's representation. Evaluations on the LibriSpeech corpus and cross-linguistic audio archive confirm the effectiveness of our proposed approach compared to the traditional ones.**

I. Introduction

Spoken term detection (STD) is defined as the task of retrieving audio segment which contains the user-defined query from audio archive. Early researches on STD mainly focus on text query task, i.e. keyword search [1], [2], which relies on automatic speech recognition (ASR) technology. Instead of using text as query input, query-by-example spoken term detection (QbE-STD) utilizes an acoustic example of query (spoken query) to detect audio segment. The input of QbE-STD task is the audio pair (spoken query and audio segment) and the output is a detection score which represents the confidence that audio segment contains the query. QbE-STD is a key technology for retrieving and browsing spoken content without transcribing them into text, which makes it attractive in the age of multimedia information explosion.

The key to solving the QbE-STD task is how to extract semantic content information while removing irrelevant information like speaker characteristics, environment noise, emotion information, etc. QbE-STD system consists of two modules: feature extraction and similarity measure. An intuitive way of QbE-STD is to directly compare the acoustic features between spoken query and audio segment, in which dynamic time warping (DTW) techniques are widely used [3], [4], [5]. In basic DTW approach, acoustic features extracted from input pair are used to construct a frame-level similarity matrix. Then the DTW algorithm is used to find the optimal warping path with the smallest distortion score through the similarity matrix. And the distortion score of the optimal warping path is returned as detection result. Improvements on DTW approach include using high-level features (phonetic posteriorgram [6], [7], bottleneck feature [8], etc.) and imposing limitations on search algorithm (segmental DTW [7], [9]). The DTW-based approach yields state-of-the-art performance in non-deep-learning approaches. However, the performance of DTW-based approach relies on high-level feature like posteriorgram feature, which is not available for low-resource audio archive. Furthermore, the dynamic programming algorithm for similarity measure is time-consuming.

Several End-to-end architectures have been proposed for QbE-STD task, which extract fixed-length vector representations from input audio pair (spoken query and audio segment), followed by a large-margin or classification training. In [10], recurrent neural networks (RNNs) with long short-term memory (LSTM) is used as encoder and the last several hidden states of RNNs are stacked to create the fixed-length vector representation. A variant using recurrent auto-encoder (RAE) is proposed in [11], in which RAE encoder is trained by margin-based loss to extract vector representation related to semantic content and then cosine distance is used for similarity measure. The parameters of RNNs used to encode query and segment are shared (i.e. the same) so that input pair is projected into the same vector space. Embedding variable-length acoustic feature sequence into fixed-length vector representation makes it easy to measure similarity. However, memory cell of RNNs maintains most information of nearby frames while losing much information of frames far away from the current time step, which indicates that representation learned by RNNs encoder depends mainly on frames at last several timesteps. We call it the position bias of RNNs encoder.

Attention mechanism has been proposed to extract representation by pooling over the whole hidden state sequence rather than retaining only several hidden states. For example, in [12], The input pair are first projected into hidden state sequences respectively by shared RNNs encoder. For spoken query, the hidden state at last timestep is used as the vector representation like [10]. As for audio segment, soft alignment scores are calculated between segment hidden state sequence and the last hidden state of query, which is then normalized to generate the attention vector. The segment hidden states are then weighted with attention vector and summed to yield the representation of audio segment. Nevertheless, this kind of attention mechanism calculates attention vector for audio segment only, which makes it a one-way attention mechanism and the asymmetric feature extraction process makes representations of query and segment less comparable.

In this paper, we present a two-way attention mechanism for feature extraction of end-to-end QbE-STD system, called attentive pooling networks [13]. The main idea is to learn a similarity measure over the projected input pair and extract information in a way that two input items can directly influence the computation of each other's representation. The two-way attention feature extractor has following advantages.

- Attention mechanism is applied to both spoken query and audio segment, which avoids position bias for both of them.
- The two-way attention feature extractor is almost symmetric for spoken query and audio segment, making representations more comparable.
- Learning a similarity measure over the projected input pair makes it possible to compare two inputs in a more plausible way, even though two inputs are not in the same semantic domain (e.g. spoken query and audio segment are from different languages) [13].

The rest of the paper is organized as follows. Section II describes in detail Shared RNNs, One-Way Attention and Two-Way Attention feature extractors. Section III introduces Two-Way Attention-based QbE-STD system. Section IV provides experiments and analyses. Finally, section V concludes the paper.

II. Two-Way Attention based Representation Learning

For notation, we denote spoken query by acoustic feature sequence $Q = \{q_1, q_2, ..., q_M\}$ and audio segment by $S = \{s_1, s_2, ..., s_N\}$. Here M and N denote the number of frames in spoken query and audio segment respectively.

A. Shared RNNs

RNNs with LSTM unit can store information for a long period of time by the means of three types of gates that control the flow of information into and out of memory cell. Given feature sequence of spoken query $Q = \{q_1, q_2, ..., q_M\}$, RNNs with LSTM unit projects $Q = \{q_1, q_2, ..., q_M\}$ into hidden state sequence $H_Q = \{h_1^Q, h_2^Q, ..., h_M^Q\}$. The hidden state at last timestep h_M^Q contains the information of the whole audio sequence and is used as vector representation of spoken query. The audio segment $S = \{s_1, s_2, ..., s_N\}$ is embedded by the same RNNs encoder into h_N^S likewise. The RNNs encoder parameters used for spoken query and audio segment are shared so that the input pair is projected into the same vector space.

The drawback of Shared RNNs encoder is that RNNs maintain most information of nearby frames while losing much information of frames far away, which makes vector representation mainly depends on the last several frames of audio sequence.

B. One-Way (OW) Attention

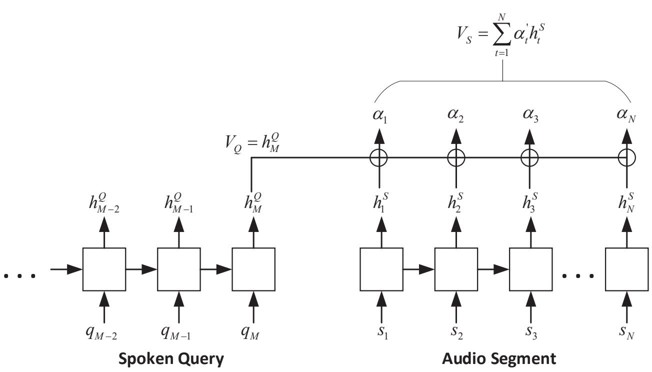

Fig. 1. One-way attention-based encoder.

To avoid the position bias of Shared RNNs encoder, attention mechanism has been proposed to pool over the whole hidden state sequence to extract information relevant to the task. The framework of one-way attention mechanism used in [12] is shown in Figure 1.

In [12], spoken query Q and audio segment S are first converted into hidden state sequences $H_Q = \{h_1^Q, h_2^Q, ..., h_M^Q\}$ and $H_S = \{h_1^S, h_2^S, ..., h_N^S\}$ by shared RNNs. For spoken query, the hidden state at last timestep h_M^Q is used as the vector representation V_Q. As for audio segment, attention value α_t at each timestep t is the cosine similarity between the query representation V_Q and the hidden state h_t^S of each frame.

$$\alpha_t = S_t \odot V_Q \tag{1}$$

where symbol $\odot$ denotes cosine similarity between vectors. Then the attention value is normalized using softmax function to get attention vector.

$$\alpha_t' = \frac{exp(\alpha_t)}{\sum_{i=1}^{N} exp(\alpha_i)} \tag{2}$$

Finally, hidden state h_t^S of audio segment at each timestep is weighted with respective normalized attention value α_t' and summed to yield the segment representation V_S.

$$V_S = \sum_{t=1}^{N} \alpha_t' h_t^S \tag{3}$$

Many more complicated attention mechanisms have been proposed in other areas; however, most of them don't apply to the QbE-STD task due to the time complexity limitation.

The attention mechanism above calculates attention vector for audio segment only, which makes it a One-Way (OW) Attention mechanism. The representation extraction of spoken query still suffers from position bias problem. Furthermore, the asymmetric feature extraction process makes extracted representations of query and segment less comparable.

C. Two-Way (TW) Attention

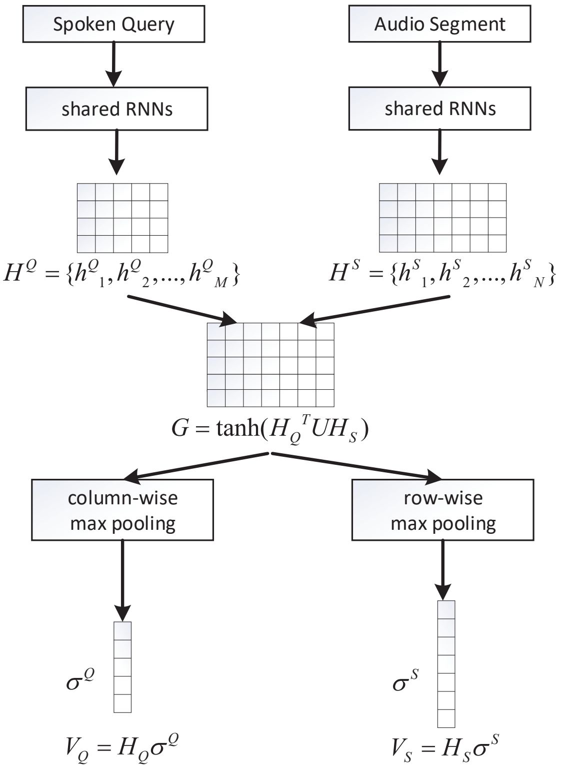

Fig. 2. Two-way attention-based encoder [13].

Here we present a Two-Way (TW) Attention mechanism, called attentive pooling networks [13] for representation learning in QbE-STD. The framework of Two-Way Attention feature extractor is shown in Figure 2.

First, RNNs with LSTM is adopted to process acoustic feature sequence of spoken query $Q = \{q_1, q_2, ..., q_M\}$ into hidden state sequence $H_Q = \{h_1^Q, h_2^Q, ..., h_M^Q\}$. And acoustic feature sequence of audio segment $S = \{s_1, s_2, ..., s_N\}$ is projected by shared RNNs into $H_S = \{h_1^S, h_2^S, ..., h_N^S\}$ likewise. Next, the attention matrix G is computed as follows.

$$G = \tanh(H_Q^T U H_S) \tag{4}$$

where U is the measure matrix and learned by training and H_Q^T is the transpose of matrix H_Q. The attention matrix G represents soft alignment score between each frame of spoken query and audio segment. Then we apply column-wise and row-wise poolings over G to generate the weight vectors $g^Q \in \mathbb{R}^M$ and $g^S \in \mathbb{R}^N$ respectively. For instance, the j-th element of the weight vectors g^Q is computed as follows.

$$[g^Q]_j = \max_{1 \leq i \leq N} [G_{j,i}] \tag{5}$$

The j-th element of the vector g^Q can be interpreted as an attention weight for the context around the j-th frame in the spoken query Q regard to audio segment S, and vice versa. Then the attention vectors g^Q and g^S are normalized with softmax function to generate attention vectors σ^Q and σ^S. Finally, the vector representations of spoken query and audio segment V_Q and V_S are computed as the dot product between the attention vector and RNNs hidden state sequence respectively.

$$V_Q = H_Q \sigma^Q, \quad V_S = H_S \sigma^S \tag{6}$$

With this design, the TW Attention mechanism can jointly learn the vector representation of input pair. The representations of query and segment are computed by pooling over the whole feature sequence with attention vector as weight, which avoids position bias of Shared RNNs encoder. Besides, TW Attention feature extractor is almost symmetric for spoken query and audio segment, which makes representations of input pair more comparable than that of OW Attention. Finally, learning a similarity measure U over the projected input pair makes it possible to compare two inputs in a more plausible way, even though two inputs are not in the same semantic domain (e.g. spoken query and audio segment are from different language) [13].

The computation process of TW Attention mechanism in [13] is actually not completely symmetric. E.g. if we exchange the position of spoken query Q and audio segment S, the measure matrix will become U^T, the transpose of U. To make TW Attention feature extractor a completely symmetric computation for query and segment, i.e. the output representations don't change if we exchange the spoken query and audio segment input, we limit the measure matrix U to a symmetric matrix, i.e. $U = U^T$.

D. Large-margin Training

We use a margin-based (hinge) loss for training, in which the intra class distance becomes smaller while the distance between classes becomes larger. For each input group during training, we construct two input pairs composed of three audio sequences: spoken query $Q = \{q_1, q_2, ..., q_M\}$, positive segment $S^{(p)} = \{s_1^{(p)}, s_2^{(p)}, ..., s_{N_1}^{(p)}\}$ (of the same word type with spoken query) and negative segment $S^{(n)} = \{s_1^{(n)}, s_2^{(n)}, ..., s_{N_2}^{(n)}\}$ (of the different word type with spoken query). Then two input pairs $(Q, S^{(p)})$ and $(Q, S^{(n)})$ are embeded into vector representations $(V_Q^{(p)}, V_S^{(p)})$ and $(V_Q^{(n)}, V_S^{(n)})$ by feature extraction module. For Shared RNNs and OW Attention encoders, $V_Q^{(p)}$ and $V_Q^{(n)}$ are the same while for TW Attention encoder they are different, because TW Attention mechanism extracts representations in a way that query and segment can influence each other's representation.

The hinge objective function is defined as follows.

$$L_{hinge} = max\{0, \quad M + l(V_Q^{(p)}, V_S^{(p)}) - l(V_Q^{(n)}, V_S^{(n)})\} \quad (7)$$

where M is the maximum distance margin between positive pair and negative pair, in this paper we set margin $M = 1$. The similarity distance l between two vector representation V_Q and V_S is computed by the cosine distance as follows.

$$l(V_Q, V_S) = (1 - cos(V_Q, V_S))/2 \quad (8)$$

III. Two-Way Attention based QbE-STD system

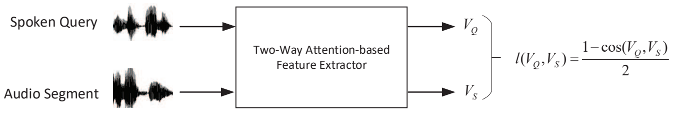
Spoken Query
Audio Segment
Two-Way Attention-based Feature Extractor
V_Q
V_S
$l(V_Q,V_S)=\frac{1-\cos(V_Q,V_S)}{2}$

Fig. 3. Framework of two-way attention-based QbE-STD system.

The fixed-length vector representation learned by TW Attention feature extractor can be applied to QbE-STD as shown in Figure 3. In off-line process, the audio archives are first segmented based on word boundaries. As for the speech of zero-resource language, voice activity detection (VAD) and skipped-frame sliding-window are available for segmentation. Then the shared RNNs part of trained encoder is used to project audio segments in archive into hidden state sequences. During on-line process, given a spoken query, the system extracts the representations of spoken query and audio segments and then rank all the audio segments in the audio archive according to the cosine distance between representations of each query and segment pair. Due to simple attention computation process and using cosine distance between single vector representations as similarity measure, the time consumption of the on-line process is very low, opposed to DTW-based approaches. For baseline systems using Shared RNNs, OW Attention as feature extractor, the framework is the same with that shown in Figure 3.

IV. Experiments

A. Experimental Setup

We use LibriSpeech [14] corpus to construct our experiment dataset. 39 dimensional MFCC acoustic features are extracted using Kaldi toolkit [15] and used as the input of all the baselines and our proposed approach.

- **Training set**: We select segments which consist of at least 6 phonemes and are of duration between 0.5 and 1.0 second from the LibriSpeech corpus. All the segments are sliced from forced aligned utterances to make sure each segment contains a complete word meaning exactly. There are 50,000 segments in the training set, which belong to 500 different word types. For each segment, we randomly select a segment of the same word type in the training set to form a positive pair and a segment of different word type to form a negative pair. An input group is formed with the two pairs above for each segment. So for each training epoch, there are 50,000 input groups (100,000 pairs) in total for training.
- **Testing set 1**: In LibriSpeech testing set, there are 100 word types, of which 50 word types don't appear in training set (Out-Of-Vocabulary, OOVs) and 50 word types are in the training set (In-Vocabulary, IVs), representing 1% positive to negative ratio to match expected application usage. For each word type, we randomly select 20 audio segments to form the audio archive for retrieving and one extra audio segment as spoken query, so there are totally 1000 OOVs audio segments and 1000 IVs audio segments in audio archive. Spoken query doesn't appear in the audio archive.
- **Testing set 2**: To evaluate the QbE-STD performance in cross-linguistic scenario, we make up an English-Chinese mixed audio archive. We select 50 Chinese word types and each word type has 20 audio segments from our private Chinese corpus. The Chinese segments are of duration between 0.5 and 1.0 second. Then we mix these 1000 Chinese audio segments with 1000 OOVs audio segments from testing set 1 to form the mixed audio archive. Each word type in the archive has one extra audio segment as spoken query. For comparison, the Chinese segmentation is based on word boundaries instead of VAD.

All the neural networks feature extractors are implemented on the TensorFlow platform. Mini-batch-trained Adam with 0.00005 learning rate and 128 batch size is used for training. RNNs in all the models consist of two hidden layers each with 128 LSTM units and all the models are trained for 3 epochs.

Mean Average Precision (MAP), the mean of the average precision in the range of recall for each query in the testing set, and P@20 are used as the evaluation metrics for QbE-STD. The approaches for the experiments are described as follows.

- **DTW-based**: the mostly used baseline for query-by-example spoken term detection. DTW baseline uses the same set of features with other end-to-end models, so they can be fairly compared.
- **Shared RNNs**: using shared RNNs as the feature extractor.
- **OW Attention**: using shared RNNs with one-way attention mechanism as the feature extractor.
- **TW Attention (Asym)**: using shared RNNs with two-way attention mechanism in [13] as the feature extractor.
- **TW Attention (Sym)**: using shared RNNs with two-way attention mechanism as the feature extractor and limiting the measure matrix U to a symmetric matrix, i.e. $U = U^T$.

B. Evaluation of QbE-STD on testing set 1 which consists of English audio segments

Table I shows the performance of all the models on the testing set 1. It's clear from the result that Shared RNNs performs better than OW Attention while DTW-based approach is worse, even though OW Attention has more parameters than Shared RNNs. We guess the reason is that the asymmetric

TABLE I
PERFORMANCE OF QBE-STD ON TESTING SET 1 WHICH CONSISTS OF ENGLISH AUDIO SEGMENTS

Model	MAP (IVs)	MAP (OOVs)	MAP (Total)	P@20
DTW-based	0.040	0.042	0.041	0.015
Shared RNNs	0.099	0.114	0.107	0.052
OW Attention	0.065	0.059	0.062	0.026
TW Attention (Asym)	**0.170**	0.131	0.151	**0.068**
TW Attention (Sym)	0.134	**0.171**	**0.153**	0.064

TABLE III
PERFORMANCE OF QBE-STD ON TESTING SET 2 WHICH CONSISTS OF ENGLISH-CHINESE MIXED AUDIO SEGMENTS

Model	MAP (Eng.)	MAP (Chi.)	MAP (Total)	P@20
DTW-based	0.047	0.040	0.043	0.018
Shared RNNs	0.161	0.101	0.131	0.052
OW Attention	0.093	0.049	0.071	0.039
TW Attention (Asym)	0.183	0.109	0.146	0.052
TW Attention (Sym)	**0.269**	**0.117**	**0.193**	**0.068**

feature extraction process for query and segment of OW Attention makes representations less comparable. We can see from the table that TW Attention (both Asym and Sym) performs the best among all the approaches and achive 140% relative MAP improvements with respect to OW Attention baseline, indicating that learning representations of input pair jointly contributes to the extraction of semantic content information. TW Attention combines the advantages of symmetric computation of Shared RNNs and attention mechanism of OW Attention.

C. Relationship between sequential phonetic structure and vector representation learned by feature extractor

TABLE II
AVERAGE COSINE DISTANCE OF THE LEARNED REPRESENTATIONS BETWEEN SEGMENTS OF INPUT PAIRS CLUSTERED BY THE PHONEME SEQUENCE EDIT DISTANCE AND WHETHER THE SUFFIXES ARE THE SAME.

Model	$D < 12$ / $D \geq 12$	Same / Different Suffixes
Shared RNNs	0.466 / 0.482	0.152 / 0.516 (+0.364)
OW Attention	0.423 / 0.457	0.402 / 0.443 (+**0.041**)
TW Attention (Asym)	0.555 / 0.590	0.525 / 0.576 (+**0.051**)
TW Attention (Sym)	0.568 / 0.596	0.533 / 0.586 (+**0.053**)

D is the phoneme sequence edit distance between segments of input pair

Table II shows the average cosine distance of the learned representations between segments of input pairs in testing set 1, on the condition of different phoneme sequence edit distance and phoneme suffixes. We select median 12 as the cosine distance boundary. It can be observed that, for all the models, cosine distance between representations of segments grows with the phoneme sequence edit distance increasing, which indicates that the learned representations can represent the sequential phonetic structure in some degree. Another observation is that the cosine distance between the learned representations of the pairs with same phoneme suffixes is less than that of pairs with different suffixes. It's worth noting that for Shared RNNs, the cosine distance of pairs with same suffixes is very small, compared to other attention-based encoders because Shared RNNs use hidden state at last timestep as the representation of input audio. Due to the elimination of position bias, attention mechanism (both OW Attention and TW Attention) can greatly reduce the difference between same and different suffixes, as compared to Shared RNNs.

D. Evaluation of QbE-STD on testing set 2 which consists of English-Chinese mixed audio segmens

Table III shows the performance of all the QbE-STD systems on testing set 2 consisting of English-Chinese mixed audio segmens. It can be seen from the result that TW Attention (both Asym and Sym) performs best among all the models, indicating that by learning a measure matrix U, TW Attention feature extractor can extract comparable semantic content information of speech from different languages, even though the second language is zero-resource. An interesting observation is that TW Attention (Sym) outperforms TW Attention (Asym), which demonstrates that by limiting the measure matrix U to a symmetric matrix and making the computation completely symmetric, TW Attention can extract more comparable representations and perform better.

V. CONCLUSIONS

In this paper, we propose a Two-Way Attention mechanism for feature extraction in QbE-STD. Evaluation on testing set consisting of English segments indicates that Two-Way Attention mechanism performs best among all the models by combining advantages of symmetric computation of Shared RNNs and attention mechanism of One-Way Attention. Besides, by learning a measure matrix, Two-Way Attention based encoder can extract comparable semantic content information of speech from different languages, even though the second language is zero-resource. Finally, limiting the measure matrix U to a symmetric matrix and making the computation completely symmetric can improve the performance of Two-Way Attention encoder in cross-linguistic scenario.

Future work includes investigating more complicated encoder neural netorks, such as TDNNs, RNNs with biLSTM, etc.

VI. ACKNOWLEDGMENTS

This work is supported by joint research fund of National Natural Science Foundation of China - Research Grant Council of Hong Kong (NSFC-RGC) (61531166002, N_CUHK404/15), National Natural Science Foundation of China (61433018, 61375027) and National Social Science Foundation of China (13&ZD189).

REFERENCES

[1] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[2] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 2004.

[3] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics Speech & Signal Processing*, vol. 26, no. 1, pp. 43–49, 2003.

[4] A. Park and J. R. Glass, "Unsupervised word acquisition from speech using pattern discovery," in *IEEE International Conference on Acoustics Speech & Signal Processing*, 2006.

[5] C. Chelba, T. J. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *Signal Processing Magazine IEEE*, vol. 25, no. 3, pp. 39–49, 2008.

[6] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 421–426.

[7] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 398–403.

[8] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Unsupervised bottleneck features for low-resource query-by-example spoken term detection." in *INTERSPEECH*, 2016, pp. 923–927.

[9] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[10] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5236–5240.

[11] Z. Zhu, Z. Wu, R. Li, H. Meng, and L. Cai, "Siamese recurrent autoencoder representation for query-by-example spoken term detection," *Proc. Interspeech 2018*, pp. 102–106, 2018.

[12] C.-W. Ao and H.-y. Lee, "Query-by-example spoken term detection using attention-based multi-hop networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6264–6268.

[13] C. d. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *arXiv preprint arXiv:1602.03609*, 2016.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.