Fast Query-by-Example Speech Search Using Attention-Based Deep Binary Embeddings

Yougen Yuan, *Student Member, IEEE*, Lei Xie, *Senior Member, IEEE*, Cheung-Chi Leung, *Member, IEEE*, Hongjie Chen, *Member, IEEE*, and Bin Ma, *Senior Member, IEEE*

***Abstract*—State-of-the-art query-by-example (QbE) speech search approaches usually use recurrent neural network (RNN) based acoustic word embeddings (AWEs) to represent variable-length speech segments with fixed-dimensional vectors, and thus simple cosine distances can be measured over the embedded vectors of both the spoken query and the search content. In this paper, we aim to improve search accuracy and speed for the AWE-based QbE approach in low-resource scenario. First, multi-head self-attentive mechanism is introduced for learning a sequence of attention weights for all time steps of RNN outputs while attending to different positions of a speech segment. Second, as the real-valued AWEs suffer from substantial computation in similarity measure, a hashing layer is adopted for learning deep binary embeddings, and thus binary pattern matching can be directly used for fast QbE speech search. The proposed approach of self-attentive deep hashing network is effectively trained with three specifically-designed objectives: a penalization term, a triplet loss, and a quantization loss. Experiments show that our approach improves the relative search speed by 8 times and mean average precision (MAP) by 18.9%, as compared with the previous best real-valued embedding approach.**

***Index Terms*—Attention mechanism, deep binary embeddings, low-resource, query-by-example, temporal context.**

I. Introduction

QUERY-BY-EXAMPLE (QbE) speech search or spoken term detection (STD) is the task of searching for spoken queries in audio archives [1], without specifying how many times a spoken query appears in each search utterance or its exact location. This task is attractive as it involves matching spoken queries with audio content directly at the acoustic level. Previous approaches usually decode the spoken query with a speech recognizer and then use a generic text-based approach for indexing [2]–[4]. Those approaches work on high-resource scenarios as they require a sizable amount of transcribed data and language-specific knowledge for system building. Resource collection at the required scale would be impossible for all 7,000 languages spoken in the world today. As for *low-resource* speech scenarios, where transcribed data and language expertise are not available, a typical approach to the task usually learns effective frame-level feature representations and then matches the spoken queries against the search content on the feature representations by dynamic time warping (DTW) [5], [6]. However, the search quality of DTW-based approaches still lags behind due to the limited discrimination capability of the frame-level feature representations. Moreover, the DTW computation itself is still inefficient for searching in a large audio collection.

As an alternative to the DTW approaches, direct matching approaches on *acoustic word embeddings* (AWEs) have recently drawn much attention for low-resource QbE speech search [7]–[13]. AWEs aim to encode variable-length speech segments into fixed-dimensional vectors with a desirable word discrimination capability. Specifically, a deep neural network (DNN) is trained using a set of spoken word pairs as weak supervision, which aims to embed both the spoken query and the search content into the same space for QbE speech search. In this way, simple vector distances (e.g., cosine) can be measured over the embedded vectors of both the spoken query and the search content. As compared with the DTW-based approaches, the AWE-based QbE speech search has reduced a large amount of computation, and the search quality has been significantly improved as well [9]–[12].

The key of the AWE-based QbE approaches is how to effectively aggregate useful information across time to fixed-dimensional discriminative vectors from a training set composed of spoken word pairs, i.e., pairs of speech segments for the same word (or word-like unit). With the recent advances in deep learning-based acoustic modeling [14]–[16], recurrent neural networks (RNNs) have been proven to be more capable of capturing temporal dependency of speech in a fixed-dimensional space, leading to state-of-the-art performances in QbE speech search [9], [13]. Specifically, deep bidirectional long short-term memory (BLSTM) RNNs have achieved superior performance with decent word discriminative ability [17].

In this paper, we propose an attention-based deep hashing network to improve search *accuracy* and *speed* for the AWE-based QbE approach in low-resource scenarios.

Our first contribution is to use multi-head self-attention to learn discriminative acoustic embeddings and improve search accuracy: Previous AWEs [12], [17], [18] are represented by the output of the last one (or several) time step(s) from the BLSTM

Manuscript received November 10, 2019; revised March 29, 2020 and May 25, 2020; accepted May 25, 2020. Date of publication May 28, 2020; date of current version July 6, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002102 and in part by the National Natural Science Foundation of China under Grant 61571363. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hsin-min Wang. *(Corresponding author: Lei Xie.)*

Yougen Yuan and Lei Xie are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: ygyuan@nwpu-aslp.org; lxie@nwpu.edu.cn).

Cheung-Chi Leung, Hongjie Chen, and Bin Ma are with the Machine Intelligence Technology, Alibaba Group, Hangzhou 311121, China (e-mail: cc.leung@alibaba-inc.com; h.chen@alibaba-inc.com; b.ma@alibaba-inc.com).

Digital Object Identifier 10.1109/TASLP.2020.2998277

2329-9290 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

network. Such kind of information aggregation in AWEs may not optimally exploit the long-range temporal context of speech. Hence we introduce an attention mechanism in our proposed network to learn more discriminative embeddings. Specifically, a multi-head self-attentive mechanism is adopted to learn a sequence of attention weights for all time steps of RNN outputs in different feature spaces (through different attention heads). Thus the essential temporal context can be fully considered, leading to more discriminative embeddings. Note that the use of attention mechanism has recently pushed the frontiers of many tasks, including neural machine translation (NMT) [19]–[21], image captioning [22], [23], speech recognition [24]–[27] and synthesis [28]–[30]. Our multi-head self-attention model can be considered as a variant of the *encoder* part of the Transformer model [21]. But one obvious difference between our model and the Transformer is the way to inject sequential information. In Transformer [21], in order to inject position information into sequence, a simple positional encoding is added to the input embeddings. In this study, BLSTM is naturally used to do positional encoding of the input speech sequence, while multi-head self-attention learns weights for different time steps of BLSTM outputs to generate more discriminative acoustic embeddings.

Our second contribution is to use deep hashing to learn binary embeddings to boost search efficiency: The learned AWEs are usually high-dimensional real-valued vectors, which still result in a substantial amount of computation in the similarity measure. Therefore, we propose to learn *deep binary embeddings* in order to significantly improve the speed of QbE speech search. The main idea is that deep binary embeddings can be calculated via binary pattern matching or Hamming distance, which can dramatically reduce the computational cost. Previous works on learning binary embeddings usually use hashing methods with an unsupervised [31]–[33] or supervised manner [34]–[36]. With the great success of deep learning, hashing methods with deep architectures have been proposed to output binary-like embeddings that preserve data structure as much as possible. Such binary embeddings have been successfully used in image retrieval [37], [38]. Especially in [39], the deep hashing network has yielded substantial computation speed-up over other hashing methods. In this paper, by introducing a hashing layer with a quantization loss in our proposed network, we convert real-valued AWEs into similarity-preserving deep binary embeddings to realize fast QbE speech search.

Our third contribution is to use three specifically-designed losses with different purposes in network training: The search accuracy and search speed are both considered in our proposed approach of attention-based deep hashing network. The network is trained to generate deep binary embeddings through three specifically-designed objectives: a penalization term, a triplet loss, and a quantization loss. The penalization term is used to force each attention head to learn dissimilar and complementary information for the final deep binary embeddings; the triplet loss makes the learned embeddings more discriminative; the quantization loss aims to reduce the error from binarizing the learned hash vectors. Finally, deep binary embeddings are obtained via binarization with a sign function. During the QbE speech search stage, a fixed-length analysis window is shifted on search content to generate deep binary embeddings, and then Hamming distances are measured over deep binary embeddings to find the audio segments that match the spoken queries.

Experiments on a low-resource QbE speech search task show that our attention-based deep binary embedding approach improves 1) the relative search speed by 8 times and 2) mean average precision (MAP) by 18.9%, as compared with the previous best real-valued embedding approach in [12]. Even in an unsupervised scenario, where the speech pairs are achieved by an unsupervised term discovery (UTD) module [40], our proposed approach still can bring 30% relative improvement on MAP over the frame-level multi-lingual bottleneck features (BNFs). We also study the effects of 1) different attention mechanisms and numbers of attention heads, 2) different quantization coefficients and numbers of bits and 3) robustness to noise interference.

To summarize our contributions, this paper proposes an attention-based deep hashing network approach that can significantly improve both search accuracy and search speed in low-resource QbE speech search. Specifically, the network is designed to minimize three losses to ensure the learned real-valued hash vectors are effective in word discrimination and the corresponding deep binary embeddings are efficient in QbE speech search.

The rest of this paper is organized as follows. Section II reviews the previous studies on both learning binary embeddings with hashing methods and QbE speech search. Section III details our proposed approach of learning deep binary embeddings via an attention-based deep hashing network for fast QbE speech search, followed by the speech search procedure in Section IV. Section V and Section VI introduce the experimental setup and results, respectively. Section VII concludes this paper and discusses our future work.

II. Related works

A. Learning Binary Embeddings With Hashing Methods

Similarity search over high-dimensional data is a common problem for various tasks, e.g., data mining, image and speech retrieval. To scale well with data dimensionality and search efficiently on a large dataset, hashing methods have been proposed for object *embeddings*, where the object can be an image, a document or a speech segment. The basic idea is to learn compact and similarity-preserving binary representations such that similar objects are mapped to nearby binary codes. With the learned compact binary embeddings, efficient pattern matching can be performed through Hamming distance. The binary embeddings can be learned in a supervised or unsupervised hashing manner. In unsupervised hashing, only unlabeled data are used to learn binary embeddings by various hash functions including locality-sensitive hashing (LSH) [31], spectral hashing (SH) [32] and iterative quantization (ITQ) [33]. In supervised hashing, labeled data are used to learn more efficient binary embeddings via various hash functions such as binary reconstruction embedding (BRE) [34], minimal loss hashing (MLH) [35] and supervised hashing with kernels (KSH) [36]. The above supervised and unsupervised hashing methods are mostly used in the image

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

retrieval task, but the performance is limited as the above hash functions are only a few linear transformations in nature. We notice that there are only two related studies in QbE speech search [9], [41], where LSH was used to replace the cosine distance, achieving substantial improvements in search accuracy and search time over the previous approaches.

As an alternative, deep learning architectures have been recently investigated for learning more efficient binary embeddings for image retrieval as they have strong non-linear modeling abilities [37]–[39] that facilitate both hash function learning and representation learning. In previous hashing methods, each input image is encoded by a vector composed of hand-crafted visual descriptors. Such hand-crafted features do not necessarily guarantee to accurately preserve the semantic similarities of each image pairs. DNNs simultaneously learn a set of hash functions as well as a useful feature representation tailored to the hashing task. Ideally, a deep hashing network learns a feature representation that sufficiently preserves the semantic similarities for images during the hashing process. Specifically, the output layer of a DNN can be simply constructed with the hash codes and the image labels during training. By feeding a test image to the trained network, we can obtain the desired hash code in the output layer. Various deep architectures, especially convolutional neural networks (CNNs) [37], [39], [42], [43], have been explored in image retrieval, resulting in reduced search time and decent retrieval accuracy.

While a large amount of labeled data are desired in the above hashing methods, low-resource scenarios have also been studied in image retrieval [44]–[46], where labeled image data are not available. In [44], the training data were augmented with different rotations of the original images and a deep CNN was trained by minimizing the binary distance between the reference image and the rotated one. In another approach [45], it sets one rotated image as a positive example and randomly selected another image as a negative example to train an unsupervised triplet hashing network. A recent study in [46] learned a hash function with a generative adversarial network (GAN). In general, although these studies have promoted the development of unsupervised methods for low-resource image retrieval, deep hashing networks, to the best of our knowledge, have not been investigated in QbE speech search.

B. QbE Speech Search

Building a low-resource QbE speech search system usually involves two steps: feature representation and similarity matching. Accordingly, the approaches can be categorized into frame-level DTW approaches and word/segment-level vector matching approaches.

In the former category, DTW is used to perform acoustic pattern matching between the spoken query and search content over the frame-level feature representations. Therefore, learning effective frame-level feature representations plays a vital role. Apart from spectral features like MFCC, more discriminative features, such as posteriorgrams, have been extracted using Gaussian mixture models (GMMs) [6], [47]–[49], deep Boltzmann machines (DBMs) [50] and acoustic segment models (ASMs) [51]–[54]. Besides, cross-lingual or multi-lingual BNFs have been frequently used in low-resource QbE speech search [55], [56], where the features are borrowed from a bottleneck-shaped DNN trained using labeled data from high-resource languages [57], [58]. Efficient frame-level feature representation can also be learned from spoken word pairs [59] discovered in the low-resource language.

Frame-level feature representations provide limited word discrimination capability and DTW-based similarity matching is highly inefficient. Recent work has shown that comparing speech segments by representing them as fixed-dimensional vectors (also known as AWEs) and measuring their vector distance can discriminate between words more accurately and efficiently. The key of AWEs is how to aggregate variable-length speech segments represented by frame-level feature sequences into compact fixed-dimensional discriminative vectors. In [8], a QbE keyword spotting approach has adopted a long short-term memory (LSTM) network for learning AWEs. The network had 15 k whole-word output targets and a fixed-length representation was extracted by choosing the last several state vectors. Note that such an approach needs a large amount of labeled speech data for learning the AWE network.

Low-resource QbE speech search has adopted the same idea but with weak supervision information instead to train the AWE network. Specifically, when a large amount of labeled data is not readily available, *weak supervision*, in the form of same or different word pairs, has been adopted. In other words, word pairs that contain different instances of the same/different word content were exploited as input to train the AWE network. Researches in this line have focused on network structures, e.g., CNNs [10], [60], RNNs [9], [12], [13] and sequence-to-sequence networks [61], [62]. On the other hand, searching in a large audio repository is time-consuming. As mentioned in Section II-A, some researchers have tried to speed up searching via a two-stage method based on LSH [9], [41].

III. Attention-Based Deep Hashing Network

A. UTD and System Overview

In low-resource scenarios, using paired examples as weak supervision has been shown to be an effective way to learn the AWE network [9], [10], [12], [60]. Word pairs are either manually labelled or discovered automatically via an unsupervised term detection (UTD) algorithm.

UTD aims to identify lexical units (e.g. word- or phrase-like units) present in the audio by searching the audio for repeated trajectories in the acoustic feature space. Specifically, each audio stream in an audio repository is first segmented into short clips and similarity measure is then conducted on these short clips. Finally similar acoustic sequences are grouped together, forming a cluster of 'discovered' lexical unit. Similar to the QbE speech search task, vast majority of UTD approaches have been built around DTW for similarity measure. In this paper, we use a popular fast implementation [40] but with two main modifications: (1) the input acoustic features in [40] are replaced with multi-lingual BNFs which represent stronger phonetic discrimination [57],

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

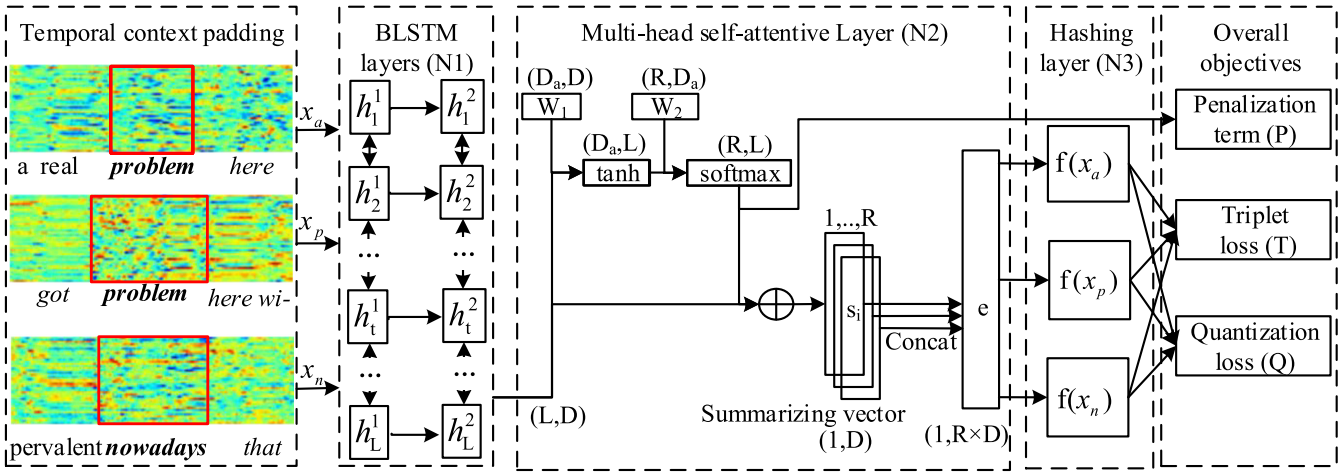

Fig. 1. The diagram of learning deep binary embeddings with a self-attentive deep hashing network.

[58]; (2) a minimum DTW similarity threshold is set to discard low-similarity speech pairs.

Fig. 1 shows the diagram of learning deep binary embeddings with a self-attentive deep hashing network. The network takes a triplet (x_p, x_a, x_n) as input. We use a pair of fixed-length speech segments as an anchor example x_a and a positive example x_p, and then randomly select another speech segment from the training set as a negative example x_n, where x_a and x_p are from the same discovered term cluster and x_n is picked from another cluster. All the triplets need temporal context padding to be fixed-length before learning the corresponding deep binary embeddings. The structure of the self-attentive deep hashing network consists of two BLSTM layers (N1), one multi-head self-attentive layer (N2) and one hashing layer (N3). This network is trained with three specifically-designed objectives including a penalization term (P), a triplet loss (T) and a quantization loss (Q). Next, we introduce the details of temporal context paddings, self-attentive deep hashing network and three specifically-designed objectives.

B. Temporal Context Paddings

The QbE speech search task matches a set of spoken queries against a search database, where the audio recordings in the database are usually much longer in duration than the queries. As word boundaries are not available, it is hard to segment the audio recordings into isolated spoken words for direct comparison. A common solution is to shift a fixed-length analysis window over the recordings to produce a sequence of fixed-length speech segments and then match the query and the speech segments with the same length, and then a simple similarity measure can be conducted on fixed-dimensional AWEs that represent the query and the fixed-length speech segments [10]. However, these fixed-length speech segments may contain a partial word unit (e.g., "wi-" in "with"), a whole word (e.g., "here"), or multiple words (e.g., "a real"), while the AWEs are usually learned using isolated spoken words (e.g., "problem" and "nowadays" in the red box in Fig. 1). Clearly there exists a mismatch between the learning of AWEs and its application on the search recordings. Our previous work [10], [12] has shown that padding temporal context during the AWE network training is useful to reduce such mismatch. Therefore, in our approach, the temporal context is also used by including the neighboring frames of each target spoken word to form speech segments with a fixed-length L.

C. Self-Attentive Deep Hashing Network

Previous AWEs [9], [12] were represented by solely merging the final hidden state vector from BLSTM networks, but such kind of information aggregation may not be able to optimally exploit the long-range context of speech. Hence in our proposed network, a multi-head self-attentive layer is added to learn a sequence of weights for all time steps of BLSTM outputs, giving different credits to different time steps. Specifically, we introduce a multi-head self-attentive mechanism [63] into the deep hashing network. Showing its superior performance in many sequence-learning tasks [21], [64]–[68], multi-head attention consists of several attention layers running in parallel, which allows the model to jointly attend to information from different representation subspaces at different positions. We expect that the use of multi-head self-attentive mechanism can bring more discriminative ability to the learned vectors. We can consider our model as a variant of the *encoder* part of the Transformer model [21], but with obvious difference. In Transformer [21], in order to inject position information into sequence, a simple positional encoding is added to the input embeddings. In this study, BLSTM naturally serves the positional encoding purpose of the input speech sequence, while multi-head self-attention learns weights for different time steps of BLSTM outputs to generate more discriminative acoustic embeddings.

Fig. 1 shows how multi-head self-attention works in our approach. We assume that the last layer N of BLSTM outputs $[h_1^N, \ldots, h_L^N]$ have the shape of $[L, D]$, where L and D represent the length of context-padded speech segments and the dimension of BLSTM outputs, respectively. The attention weight matrix A

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

is calculated by

$$A = \text{softmax}(W_2 \tanh(W_1 H^T)), \tag{1}$$

where H^T is the transpose of BLSTM outputs $[h_1^N, \ldots, h_L^N]$. W_1 and W_2 are the matrices of size $[D_a, D]$ and $[R, D_a]$, respectively. D_a and R represent the dimension of attention weights and the number of attention heads, respectively. The activation function softmax() ensures that all the attention weights sum up to 1 on each attention head. For the i-th attention head, the summarizing vector s_i is calculated by

$$s_i = \sum_{t=1}^{L} A_{it} h_t^N, \tag{2}$$

where the attention weights A_{it} represents the attention weight matrix A at the attention head i and the time step t. Then, all the summarizing vectors are unfolded to form the resulting embedding, which can be calculated as

$$e = [s_1, \ldots s_i, \ldots, s_R], \tag{3}$$

where [] represents the vector concatenation operation, R is the number of attention heads. By using multiple attention heads, the attention mechanism is subject to attend diverse information from different representation subspaces.

However, the above embeddings are usually a real-valued high-dimensional vector. Similarity measure over such vectors needs a large amount of computation during QbE speech search. Hence in our proposed network, a hashing layer is added to compress the real-valued high-dimensional vector into a K-dimensional hash vector $f(x)$ that consists of a fully connected layer with K hidden units. If the concatenation operation is used, the hash vectors $f(x)$ are calculated as

$$f(x) = \tanh(We + b) \tag{4}$$

where $\tanh$ is the hyperbolic tangent activation function to squash the results to be within $[-1, 1]$. Finally, the binary embeddings $b(x)$ can be obtained from the hash vectors $f(x)$ with the sign function $sgn()$, which are calculated by

$$b(x) = \begin{cases} 0 & \text{if } sgn(f(x)) = -1 \\ 1 & \text{if } sgn(f(x)) = 1 \end{cases}. \tag{5}$$

D. Three Specifically-Designed Objectives

To encourage diversity between any two summarizing vectors, a penalization term P is introduced when $R > 1$ as

$$P = \| AA^T - I \|_2^F, \tag{6}$$

where I is an identity matrix and $\| \cdot \|^F$ represents the Frobenius norm. With the penalization term P, we expect different hash vectors $f(x)$ to contain more dissimilar information from each other, representing different feature representations for the same speech segment.

Besides, to learn more discriminative hash vectors $f(x)$, a triplet loss is employed during the training of a self-attentive deep hashing network. The triplet loss aims to decrease the distance between the hash vectors $f(x_p)$ and $f(x_a)$, while increase the distance between the hash vectors $f(x_n)$ and $f(x_a)$ at the same time. Specifically, the triplet loss is defined as

$$T = \max(0, \theta + D_c(f(x_p), f(x_a)) - D_c(f(x_n), f(x_a))), \tag{7}$$

where θ is a constant margin that regularizes the gap between the cosine distance of same-word pairs $D_c(f(x_p), f(x_a))$ and the cosine distance of different-word pairs $D_c(f(x_n), f(x_a))$.

Last but not least, a quantization loss is added to quantize the hash vector $f(x)$ to binary-like embeddings. The quantization loss aims to transform the elements in the hash vector $f(x)$ as close to -1 and 1 as possible. Specifically, the quantization loss is defined as

$$Q = \sum_{x \in (x_a, x_p, x_n)} \| \, | f(x) | - 1 \, \|_1, \tag{8}$$

where $1 \in R^K$ is the vector of ones. With the quantization loss Q, we are easy to reduce the gap between real-valued hash vectors $f(x)$ and deep binary embeddings $b(x)$.

By joint-training the above three specifically-designed objectives, the overall objectives O is calculated as

$$O = \alpha P + \beta T + \gamma Q, \tag{9}$$

where $\alpha/\beta/\gamma$ are coefficients of the penalization term P in Eq.(6), the triplet loss T in Eq.(7), and the quantization loss Q in Eq.(8), respectively.

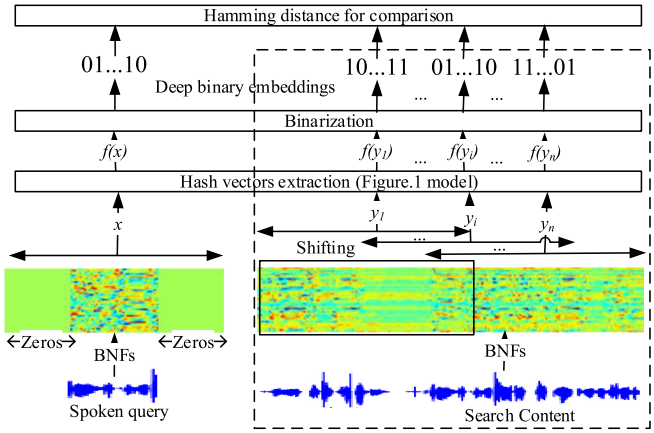

Fig. 2. The process of embeddings-based QbE speech search. A Hamming distance is performed between the spoken query and search content over deep binary embeddings.

IV. Embeddings-Based Fast QbE Speech Search

Fig. 2 illustrates the process of fast QbE speech search using the deep binary embeddings. Firstly, a fixed-length analysis window (the black box in Fig. 2) is shifted on the search content y along the time axis to obtain a sequence of hash vectors $(f(y_1), \ldots, f(y_i), \ldots, f(y_n))$ via the trained self-attentive deep hashing network. Then, binarization in Eq.(5) is conducted to convert the real-valued hash vectors into deep binary embeddings $(b(y_1), \ldots, b(y_i), \ldots, b(y_n))$. Both operations (the black dashed box of Fig. 2) can be calculated in advance to save search run-time. Following [10] and [12], as no context information

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

is available in the spoken query x, zeros are firstly padded on both sides of x to the same length as the analysis window, and then the spoken query x is also converted into deep binary embeddings $b(x)$. Next, Hamming distance is measured over the deep binary embeddings between the spoken query and the search content. Note that using Hamming distance over binary embeddings greatly reduces computation as compared with using cosine distance over real-valued embeddings. Hence our proposed approach facilitates fast QbE speech search.

With the deep binary embeddings of the spoken query $b(x)$ and the deep binary embeddings of the i-th target speech segment $b(y_i)$, the Hamming distance is calculated by

$$D_h(b(x), b(y_i)) = \frac{1}{K} \sum_{k=0}^{K-1} \mid b(x)_k - b(y_i)_k \mid \qquad (10)$$

where K is the total number of elements in the deep binary embeddings. Suppose the search content y contains n speech segments $[y_1, \ldots, y_i, \ldots, y_n]$, the cost between the spoken query x and the search content y can be calculated by

$$Cost(x, y) = \min\{D_h(b(x), b(y_i))\}, i = 1, \ldots, n \qquad (11)$$

Finally, given a spoken query, the costs of all search content in the speech database are returned according to Equation 11 for ranking.

V. Experimental Setup

A. Datasets and BNFs With Temporal Context Padding

We have carried out experiments to evaluate the proposed approach. In this paper, English is considered as the low-resource target language and QbE speech search is conducted on the English switchboard telephone speech corpus (LDC97S62). Following the previous work [10], [59], we regard Mandarin and Spanish as the resource-rich non-target languages in a multi-lingual BNF extraction network. Specifically, the network is trained using 170 hours of labeled data from the HKUST Mandarin telephone speech corpus (LDC2005S15) and 152 hours of labeled data from the fisher Spanish telephone speech corpus (LDC2010S01). Taking 39-dimensional filter-banks with pitch as input, the network has four hidden layers, where the third layer serves as a bottleneck with 40 hidden units and the other three layers have 1500 hidden units per layer. Notice that all the hidden layers are shared by different languages, while the last softmax output layer is associated with the triphone states of each language. Specifically In Mandarin and Spanish, there are 412 and 420 tied triphone states, respectively. Subsequently, in the target language, i.e., English, the speech data is represented by 40-dimensional multi-lingual BNFs extracted from the trained network.

We use the same set of spoken word pairs as in [12] for a fair comparison. Note that the spoken word pairs are manually created as weak supervision in learning AWE. Table I summarizes the statistics of each dataset. The training set has the vocabulary size (the number of unique spoken words) of 1.6 k and it includes 7.9 k spoken word instances. At the same time, 11 k spoken word instances with the vocabulary size of 3.9 k are used as the development set. Each spoken word instance consists of at least 5 characters and has a duration between 0.5 to 2 seconds. Before training the embedding network, we set each target spoken word instance in the middle and pad its original temporal context on both sides to form a speech segment with 2 seconds. All the spoken word instances in the training set can make up about 10 k same word pairs for learning the AWE network.

TABLE I
The Statistics of Learning Embeddings for QbE Speech Search

Datasets	#Vocabulary	#Spoken word instances	#Same word pairs
Training set	1.6k	7.9k	10k
Development set	3.9k	11k	-

As for the QbE speech search test, the keyword set is composed of 346 spoken keyword queries. Each keyword query is a single word which includes at least 5 characters and has a duration between 0.5 to 2 seconds. Meanwhile, the search content contains 10 hours of speech from the Switchboard telephone speech corpus. The keyword set is not overlapped with the training set for fair evaluation. Note that the QbE speech search task can handle spoken queries containing multiple words or phrases. As our aim is to evaluate the quality of acoustic embeddings, only single word queries are considered, same as the common setup in the literature [51], [69].

B. Unsupervised Term Detection

In an unsupervised scenario, word pairs are not available so that they have to be discovered automatically from scratch in order to train the AWE network. Hence we also examine the performance of the proposed approach in such an unsupervised scenario. Instead of using the above manual set in Section V-A, a UTD module is used to discover the word-like speech pairs in an unsupervised manner. Here we use the open-source UTD tool ZRTools,[1] which is based on DTW to identify repeated speech segments in an unlabeled corpus [40]. We have observed that the UTD process requires more data to discover the same amount of word-like speech pairs as the manual training set in Section V-A. Thus we use about 37 k unlabeled spoken word instances from an extended training set (Set 1 in [10], [12]). Besides, 40-dimensional multi-lingual BNFs are used as input in the UTD process. As for the post-processing of UTD, a minimum DTW similarity threshold is set to 0.91 empirically to discard low-similarity word-like speech pairs.

C. Network Details

The self-attentive deep hashing network is trained using the manual or discovered spoken word pairs, and then the QbE speech search is performed on the learned deep binary embeddings. The deep hashing network consists of two BLSTM layers, one multi-head self-attentive layer, and one hashing layer. Each BLSTM layer consists of 512 hidden units on both forward and backward directions, respectively. For the self-attentive layer,

[1] [Online]. Available: https://github.com/arenjansen/ZRTools

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISON OF DIFFERENT WORD-LEVEL EMBEDDINGS FOR QBE SPEECH SEARCH

System	Embeddings	Temporal context	Network layers	Overall objectives	Binarization	Similarity measure	QbE speech search			
							MAP	P@N	P@5	Time(s)
S1	BLSTM	No	N1	T	No	cosine	0.464	0.424	0.503	717.0
S2					Yes	Hamming	0.415	0.379	0.450	89.0
S3	BLSTM [12]	Yes	N1	T	No	cosine	0.481	0.441	0.512	718.0
S4					Yes	Hamming	0.428	0.392	0.472	89.5
S5	BLSTM+Hashing	Yes	N1+N3	T+Q	No	cosine	0.483	0.443	0.530	718.0
S6					Yes	Hamming	0.472	0.437	0.513	90.0
S7	BLSTM+Attention	Yes	N1+N2	T+P	No	cosine	**0.583**	0.533	**0.677**	716.0
S8					Yes	Hamming	0.552	0.507	0.616	89.4
S9	BLSTM+Attention+Hashing	Yes	N1+N2+N3	T+P+Q	No	cosine	0.564	0.527	0.617	716.0
S10					Yes	Hamming	0.572	**0.535**	0.639	**89.2**

the dimension of attention weights D_a and the number of attention heads R are set to 320 and 5, respectively. At the last hashing layer, the dimensions K of hash vectors $f(x)$ are set to 1,024. As for the objective function in Eq.(9), we empirically set the coefficients α, β and γ to 0.01, 1 and 0.01, respectively. The three coefficients ($\alpha/\beta/\gamma$) aim to keep the loss reduction of the three objectives ($P/T/Q$) at the same order of magnitude. The ordinary BLSTM network for comparison consists of two BLSTM layers with 512 hidden units per direction and per layer. After training, binarization is also performed to convert the learned hash vectors $f(x)$ into deep binary embeddings $b(x)$ via Eq. 5.

The initial weights of the embedding networks are randomly set from -0.05 to 0.05. An Adam optimizer [70] is used for updating the weights with the mini-batch size of 100 and the initial learning of 0.0001. A dropout rate of 0.4 is set to both BLSTM layers. All the embedding networks are implemented using the Tensorflow toolkit [71].

D. Evaluation

The performance of QbE speech search is evaluated by three metrics, including mean average precision (MAP), the precision of the top N utterances (P@N), and the precision of the top 5 utterances (P@5). The precision of the top K search content is calculated by

$$P@K = \frac{1}{M}\sum_{i=1}^{M}\frac{\sum_{j=1}^{K} hit(q_i, j)}{K} \tag{12}$$

where M denotes the number of spoken queries. $hit(q_i, j) \in \{0, 1\}$ with 1 represents that the j-th ranked utterance contains the spoken query q_i and 0 otherwise. Here we report $P@5$ and $P@N$. We also report mean average precision (MAP) by

$$MAP = \frac{1}{N}\sum_{K=1}^{N} P@K, \tag{13}$$

where N is the number of utterances containing the query term. High precision represents better performance.

In addition, the time of calculating the minimum cost via the Eq. 11 is tested between all the spoken queries and search content over the learned AWEs. All the tests are performed using a computation thread on a workstation equipped with an Intel Xeon E5-2680 @ 2.7 GHz CPU.

VI. EXPERIMENTAL RESULTS

A. Comparison of Different Embeddings

Table II summarizes the results of QbE speech search using different word-level embeddings. Here the BLSTM embeddings [12] are served as the baseline as they have achieved the best performance in recent QbE speech search. Building upon the BLSTM, we add the temporal context on each target spoken word, the multi-head self-attentive layer (denoted as Attention) and the hashing layer (denoted as Hashing). Fig. 1 shows the detailed configuration of the network layers (N1/N2/N3) and the objective functions (P/T/Q). Binarization is the process of converting real-valued embeddings into the corresponding binary embeddings via Eq.(5). During the similarity measure, cosine and Hamming distances are used for real-valued and binary embeddings, respectively. For a fair comparison, all these embedding networks are trained using the same training set with 10 k spoken word pairs represented by multi-lingual BNFs.

We first investigate the results before binarization. The BLSTM system trained with temporal context (S3) performs better than that trained without temporal context (S1). This result demonstrates that using temporal context padding plays an important role in learning effective embeddings for QbE speech search. Besides, we notice that S5 achieves comparable performance with S3 for QbE speech search. This indicates that using hashing layer and quantization loss does not affect the searching accuracy. With the help of the attention mechanism, S7 achieves significant performance improvement as compared with the baseline BLSTM system (S3). The combination system (S9), although with a small search quality degradation, finally brings a large relative improvement of 17.2%/19.5%/20.5% in MAP/P@N/P@5 respectively, as compared with S3. These results suggest that both temporal context and attention mechanism are critical of learning more discriminative embeddings for QbE speech search.

By applying binarization on the learned real-valued embeddings, the resulting binary embeddings (S2/S4/S6/S8/S10) improve the relative search speed by about 8 times compared with the real-valued vectors (S1/S3/S5/S7/S9). This indicates that using Hamming distance on deep binary embeddings requires much lower computation than using cosine distance on real-valued embeddings. If binarization is directly applied to the real-valued vectors without quantization (comparing S1 and S2, S3 and S4, S7 and S8), although the search speed has the

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

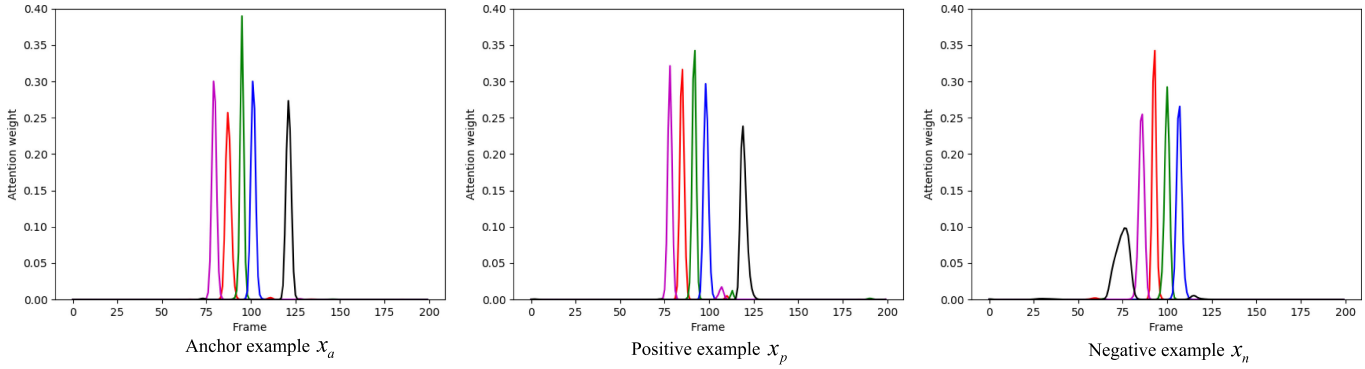

Fig. 3. Visualization the attention weights of multi-head self-attentive mechanism from one triplet in the training set. Each colored line represents the attention weights of one head.

same level of improvement, the search accuracy of the resulting binary embeddings suffers from clear performance degradation. As a comparison, when a hashing layer with a quantization loss is performed (S6/S10), the search accuracy maintains at the same level, meanwhile with about 8 times speed-up in search time. This result demonstrates the superiority of quantization in learning deep binary embeddings. The quantization can effectively reduce the binarization error and lead to nearly lossless embeddings for QbE speech search. Our conclusions in speech search are in line with using binary embeddings in image retrieval [39].

In summary, when both attention and hashing layers are considered on training a deep hashing network with temporal context padding (S10), the resulting deep binary embeddings achieve a superior tradeoff between search accuracy and search speed in QbE speech search. Compared with the previous best BLSTM embedding system (S3), S10 not only speeds up the search computation by 8 times but also improves the search accuracy by 18.9%/21.3%/24.8% in terms of MAP/P@N/P@5, respectively.

B. Investigation on Multi-Head Self-Attentive Mechanism

We investigate the effect of multi-head self-attentive mechanism in learning deep binary embeddings for QbE speech search. Experiments are carried on a BLSTM system with a multi-head self-attentive mechanism (S8). Here we still use the 10 k spoken word pairs represented by multi-lingual BNFs in the training set for network training. Fig. 4 shows the effect of different attention head in learning self-attentive deep binary embeddings for fast QbE speech search. We can find that when the number of attention heads is increased from 1 to 5, the learned self-attentive deep binary embeddings provide a better search accuracy. But when we continue to increase the number to 9, the search accuracy is dropped. These results suggest that using a 5-head self-attentive mechanism learns the most effective deep binary embeddings for QbE speech search.

We also investigate the importance of the penalization term (P). For an ablation study, we train a 5-head self-attentive network without P. From Table III, we find that the penalization term is essential for a better performance. Without P (system

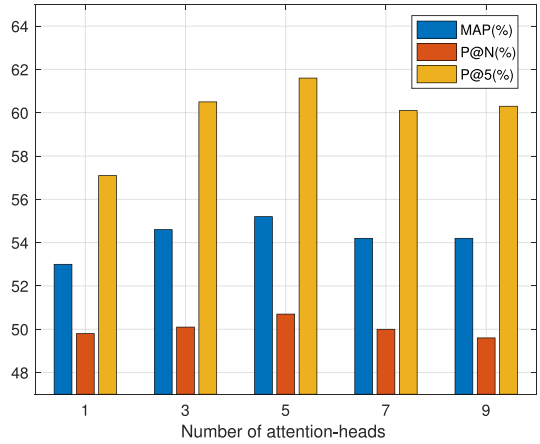

Fig. 4. Effect of different numbers of attention-heads in learning attention-based deep binary embeddings for fast QbE speech search.

TABLE III
EFFECT OF PENALIZATION TERM IN LEARNING SELF-ATTENTIVE DEEP BINARY EMBEDDINGS FOR FAST QBE SPEECH SEARCH. NUMBER OF ATTENTION HEADS IS SET TO 5

System	Objective	QbE speech search		
		MAP	P@N	P@5
S8	T+P	**0.552**	**0.507**	**0.616**
S8F	T	0.542	0.498	0.601

S8F), the MAP degrades from 0.552 (S8) to 0.542. This may indicate that the penalization is beneficial for encouraging the diversity of summation weight vectors across different hops of attention.

We further visualize the multi-head attention weights of one triplet (x_p, x_a, x_n) in the training set. Note that anchor example x_a contains the same target word with the positive example x_p while a different target word with the negative example x_n. In Fig. 3, the horizontal and vertical coordinates represent the number of frames of the example and the attention weights, respectively. Different heads in the multi-head self-attentive mechanism are represented by different colors. First of all, there is almost no overlap between any two attention heads to every example. This observation verifies the multi-head self-attentive

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
EFFECT OF DIFFERENT QUANTIZATION COEFFICIENTS IN THE LEARNING OF DEEP BINARY EMBEDDINGS FOR FAST QBE SPEECH SEARCH

System	Quantization coefficient	QbE speech search		
		MAP	P@N	P@5
S8	0.000	0.552	0.507	0.616
S8H	0.001	0.563	0.515	0.620
S10	**0.010**	**0.572**	**0.535**	**0.639**
S8I	0.100	0.451	0.414	0.484

TABLE V
COMPARISON BETWEEN ORACLE MULTI-HEAD ATTENTION (MHA) AND THE PROPOSED IN LEARNING DEEP BINARY EMBEDDINGS FOR FAST QBE SPEECH SEARCH

System	Attention mechanism	#Model para. (M)	QbE speech search		
			MAP	P@N	P@5
S8G	Oracle MHA	5.06	0.552	0.517	0.602
S10	Proposed	4.50	**0.572**	**0.535**	**0.639**

mechanism is capable of attending to different positions of a sequence for joint representation. More importantly, the weight patterns between the anchor example x_a and the positive example x_p are very similar, while the weight pattern of the negative example x_n is significantly different from the two. We find similar observations from many other triplets. These observations suggest that the multi-head self-attentive mechanism can learn effective information aggregation, which results in discriminative deep binary embeddings.

C. Comparison With Oracle Multi-Head Attention

As mentioned in Section I, our multi-head self-attention model can be considered as a variant of the encoder part of the Transformer model [21] which also uses multi-head attention but in a different way. We make a simple comparison between the way used in this paper and the oracle multi-head attention (MHA). Specifically, given the outputs of the last BLSTM layer, the oracle MHA firstly projects the BLSTM outputs to three different matrices (Query, Key and Value), and it takes all three matrices to the attention mechanism to produce the corresponding output vectors. Then, all the vectors are concatenated and fed into the next hashing layer. For a fair comparison, in our experiments, the number of heads in the oracle MHA is also set to 5 and all three matrices have dimensions of 1024. We list the QbE speech search results in Table V. We can find that the multi-head self-attentive mechanism used in this paper not only has a better performance, but also makes the whole deep hashing network have fewer parameters. The performance of oracle MHA is also good which shows that the attention mechanism is beneficial to learning effective embeddings for QbE speech search.

D. Deep Binary Embeddings With Different Quantization Coefficients

From the integrated objective function in Eq.(9), we can find that there is a coefficient γ added on the quantization loss to control the quantization speed. Varying of the speech-controlling coefficient may result in a difference in the discrimination capability of the deep binary embeddings. The comparison results are shown in Table IV. Here the experiments are based on S8 and S10. We can see that when the quantization coefficient is increased from 0 (no quantization) to 0.01, more effective deep binary embeddings can be learned. However, when we keep increasing the quantization coefficient to another magnitude (0.1), the performance of the learned deep binary embeddings suffers from a clear degradation. Although a higher quantization coefficient can speed up the quantization process, the learned binary embeddings are hard to obtain the same level of discriminative capability as a lower quantization coefficient. Therefore, setting a proper quantization coefficient is also critical of learning deep binary embeddings for QbE speech search. In other words, we need to balance the quantization speed and the discriminative capability learned from the embeddings.

E. Deep Binary Embeddings With Different Numbers of Hashing Bits

It is meaningful to investigate how many hashing bits the deep binary embeddings can be compressed to without obvious performance degradation. Here we vary the bits of $[32, 64, 128, 256, 512, 1024, 2048]$ and test performance of the learned deep binary embeddings for QbE speech search. The search time is also recorded for comparison. From Table VII, we can find that when the number of bits is increased from 32 to 128, the learned deep binary embeddings provide a better search accuracy. But when we increase the number to 512, the search accuracy only has an ignorable gain while the search time is significantly increased. When the number of bits is continually increased to 1024, the learned deep binary embeddings hold the best search accuracy. Finally, when the number of bits is set to 2048, the search accuracy suffers from a clear drop and the search time faces a huge increase. In summary, we should learn deep binary embeddings with a suitable number of bits to balance search accuracy and time.

F. Effect of Spoken Queries on Noise Interference

In real applications, spoken queries or search contents may be contaminated by noise or channel interference. This will induce a mismatch problem. Hence we further investigate the robustness of the proposed approach to noise interference on the spoken queries. To obtain the noisy spoken queries during the testing phase, we manually add white Gaussian noise to the original spoken queries at the different signal-to-noise-ratio (SNR), including 0 dB, 5 dB, 10 dB, 15 dB and 20 dB. In order to simulate training-testing mismatch, the embeddings of the noisy spoken queries are then extracted from the network that is still trained with the clean speech segments. TABLE VIII lists the results for QbE speech search at different noise conditions of spoken queries. Here we compare our proposed system (S10) with the previous state-of-the-art system (S4). Not surprisingly, as the noise interference increases in spoken queries, the performance of both systems (S4 and S10) becomes worse. Most importantly, no matter at what level of noise interference, our proposed approach (S10) consistently performs much better than the previous approach (S4). This indicates that our proposed acoustic word embedding is more robust to noise interference.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
EFFECT OF DEEP BINARY EMBEDDINGS USING DISCOVERED SPEECH PAIRS. THE UTD IS PERFORMED ON THE EXTENDED TRAINING SET

System	Representation	Speech pairs	Post-processing of UTD					QbE speech search		
			Minimum DTW threshold	#Classes	#Speech segments	#Speech pairs	Pair's accuracy	MAP	P@N	P@5
S11	Baseline (Multi-lingual BNFs) [59]	N/A	N/A	N/A	N/A	N/A	N/A	0.400	0.365	0.485
S12A	Deep binary embeddings based on S10	UTD	0.89	685	27,430	172,178	0.821	0.442	0.405	0.501
S12B			0.90	1,094	24,862	120,264	0.874	0.492	0.455	0.549
S12C			**0.91**	1,674	20,575	68,720	0.909	**0.520**	**0.477**	**0.590**
S12D			0.92	2,121	14,503	29,750	0.931	0.452	0.415	0.511
S12E			0.93	2,000	8,168	9,234	0.946	0.325	0.291	0.359
S13	Topline (deep binary embeddings based on S10)	Ground truth	N/A	1,687	20,575	68,720	1.0	0.617	0.570	0.662

TABLE VII
EFFECT OF DIFFERENT NUMBERS OF HASHING BITS IN THE LEARNING OF DEEP BINARY EMBEDDINGS FOR FAST QBE SPEECH SEARCH

System	#hashing bits	QbE speech search			
		MAP	P@N	P@5	Time(s)
S10A	32	0.422	0.394	0.434	3.4
S10B	64	0.490	0.457	0.522	6.7
S10C	128	0.524	0.489	0.575	12.2
S10D	256	0.526	0.483	0.572	18.8
S10E	512	0.530	0.490	0.586	44.6
S10	**1,024**	**0.572**	**0.535**	**0.639**	89.2
S10G	2,048	0.540	0.495	0.595	161.5

TABLE VIII
QBE SPEECH SEARCH RESULTS FOR TESTING QUERIES AT DIFFERENCE SNR CONDITIONS

Conditions of spoken queries	S4			S10		
	MAP	P@N	P@5	MAP	P@N	P@5
Clean	0.428	0.392	0.472	0.572	0.535	0.639
SNR=20dB	0.414	0.383	0.461	0.551	0.512	0.605
SNR=15dB	0.389	0.356	0.429	0.513	0.475	0.564
SNR=10dB	0.359	0.329	0.380	0.464	0.427	0.493
SNR=5dB	0.315	0.282	0.334	0.404	0.374	0.407
SNR=0dB	0.266	0.233	0.248	0.338	0.305	0.316

G. Deep Binary Embeddings Using Discovered Speech Pairs From UTD

We also examine the performance of our proposed deep hashing network system in an unsupervised scenario where the word pairs are not available. Specifically, a UTD module described in Section V-B is performed on the extended training set to discover high-quality word-like speech pairs, and then deep binary embeddings can be learned using the speech pairs based on the self-attentive deep hashing network (S10). In the post-processing of UTD, the minimum DTW similarity threshold controls the number and precision of the discovered word pairs. Table VI summarizes the accuracy of QbE speech search when a different threshold is used in the UTD post-processing. When the minimum DTW similarity threshold varies, we obtain different numbers of clustered groups (#Classes), discovered speech segments (#Speech segments), and discovered speech pairs (#Speech pairs). Subsequently, the accuracy of speech pairs (pair's accuracy) also changes. It means the quality of word-like speech pairs discovered from UTD is different. Besides, we use the results of multi-lingual BNFs as the baseline (S11). For a fair comparison, we are particularly interested in the deep embedding network (S12 C) trained using the same number of speech pairs with the Topline system (S13) that uses the ground truth speech pairs manually created.

When the threshold is set to 0.91 (S12 C), we notice that the learned deep binary embeddings trained with the discovered speech pairs hold the best performance in QbE speech search. Compared with the multi-lingual BNFs, the relative improvement in MAP/P@N/P@5 can be up to 30.0%/30.7%/21.6% respectively. Such significant improvement demonstrates that the proposed approach is also effective in an untranscribed speech scenario. At the same time, the deep binary embeddings learned from fully untranscribed speech still have a clear performance gap with those embeddings trained using the provided manual speech pairs.

From Table VI we can see that when the minimum DTW similarity threshold is increased, the discovered spoken words are restricted to those with higher similarity. At the same time, the number of speech segments and the number of word pairs are decreasing while the accuracy of the discovered speech pairs is rising. When the threshold is increased from 0.89 to 0.91, the resulting deep binary embeddings bring better accuracy in speech search. When the threshold is increased to $\geq$0.92, the number of discovered speech pairs becomes fewer although more accurate speech pairs are discovered. The results suggest that these speech pairs are not enough to train good deep binary embeddings that facilitate QbE speech search. Another limitation of setting a high DTW threshold is that the same-word speech pairs are probably clustered to different classes so that the errors of choosing negative examples will increase in training the deep embedding network.

We finally investigate how the training data size in UTD affects the speech search quality. As mentioned in Section V-B, in experiments above, we adopt an extended training set (Set 1 in [10], [12]) to augment the training data in order to detect more spoken word instances for AWE network training. Now we remove this extra training set to see how fewer spoken word instances affect the AWE network training and the speech search performance. From Table IX, we can find that a larger training set in UTD is essential for finding more word-like pairs and boosting the final QbE speech search performance. Performance degrades significantly when only using the original training set (with small amount of word-like pairs) to train the deep binary embedding network, although S14 C still outperforms the multi-lingual BNFs baseline (S11) in terms of MAP and P@N.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

TABLE IX
EFFECT OF TRAINING DATA SIZE IN UTD FOR FAST QBE SPEECH SEARCH

System	Representation	Speech pairs	Post-processing of UTD					QbE speech search		
			Minimum DTW threshold	#Classes	#Speech segments	#Speech pairs	Pair's accuracy	MAP	P@N	P@5
S11	Multi-lingual BNFs [59]	-	-	-	-	-	-	0.400	0.365	0.485
S14C	Deep binary embeddings based on S10	UTD on training set only	0.91	658	5,005	12,361	0.673	0.432	0.399	0.476
S12C	Deep binary embeddings based on S10	UTD on extended training set	0.91	1,674	20,575	68,720	0.909	**0.520**	**0.477**	**0.590**

VII. CONCLUSION

We have proposed a novel approach to improve search accuracy and search speed for QbE speech search. The approach of learning deep binary embeddings from a self-attentive deep hashing network significantly outperforms the previous state-of-the-art BLSTM embedding system in terms of both search accuracy and search efficiency. More precisely, the introduction of temporal context padding and multi-head self-attentive mechanism in the network significantly improves search accuracy, and the introduction of hashing layer with quantization loss largely reduces search time while maintaining the performance. We have also conducted extensive experiments on the analysis of the multi-head self-attentive mechanism, the deep hashing network with different quantization coefficients and the number of bits, and the spoken queries in different noise conditions. Experimental results have demonstrated the superior performance of our proposed approach in both search accuracy and search speed. In the future, as there are many recent new advances in attention mechanism [27], [30], [72], we will examine their abilities in learning acoustic word embeddings and further improve search quality and robustness. We will also apply our embedding network to other low-resource speech applications.

REFERENCES

[1] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 186–197, 2008.

[2] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, "A lattice-based approach to query-by-example spoken document retrieval," in *Proc. SIGIR*. ACM, 2008, pp. 363–370.

[3] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *Proc. Autom. Speech Recognit. Understanding*, 2009, pp. 404–409.

[4] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, "Spoken content retrieval: Beyond cascading speech recognition with text retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 9, pp. 1389–1420, 2015.

[5] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. Autom. Speech Recognit. Understanding*, 2009, pp. 421–426.

[6] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. Autom. Speech Recognit. Understanding*, 2009, pp. 398–403.

[7] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. Autom. Speech Recognit. Understanding*, 2013, pp. 410–415.

[8] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5236–5240.

[9] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proc. INTERSPEECH*, 2017, pp. 2874–2878.

[10] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Learning acoustic word embeddings using paired examples for query-by-example speech search," in *Proc. INTERSPEECH*, 2018, pp. 97–101.

[11] C.-W. Ao and H.-y. Lee, "Query-by-example spoken term detection using attention-based multi-hop networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 6264–6268.

[12] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Query-by-example speech search using recurrent neural acoustic word embeddings with temporal context," *IEEE Access*, vol. 7, no. 1, pp. 67 656–67 665, 2019.

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

[14] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6645–6649.

[15] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. Autom. Speech Recognit. Understanding*, 2013, pp. 273–278.

[16] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[17] S. Settle and K. Livescu, "Discriminative acoustic word embeddings: Tecurrent neural network-based approaches," in *Proc. SLT*, 2016, pp. 503–510.

[18] W. He, W. Wang, and K. Livescu, "Multi-view recurrent neural acoustic word embeddings," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–12.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[20] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[21] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[22] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[23] L. Li, S. Tang, L. Deng, Y. Zhang, and Q. Tian, "Image caption with global-local attention," in *Proc. Assoc. Adv. Artif. Intell.*, 2017, pp. 4133–4139.

[24] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[25] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[26] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4945–4949.

[27] C.-C. Chiu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4774–4778.

[28] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. INTERSPEECH*, 2017, pp. 4006–4010.

[29] J. Shen *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4779–4783.

[30] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural speech synthesis with transformer network," in *Proc. Assoc. Adv. Artif. Intell.*, 2019.

[31] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *Proc. Very Large Data Bases*, 1999, pp. 518–529.

[32] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[33] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, 2013.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

[34] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.
[35] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 353–360.
[36] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 2074–2081.
[37] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. Assoc. Adv. Artif. Intell.*, 2014, pp. 2–8.
[38] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3270–3278.
[39] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. Assoc. Adv. Artif. Intell.*, 2016, pp. 2415–2421.
[40] A. Jansen and B. V. Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. Autom. Speech Recognit. Understanding*, 2011, pp. 401–406.
[41] K. Levin, A. Jansen, and B. V. Durme, "Segmental acoustic indexing for zero resource keyword search," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5828–5832.
[42] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval," in *Proc. Assoc. Adv. Artif. Intell.*, 2016, pp. 3457–3463.
[43] Y. Cao, M. Long, J. Wang, and S. Liu, "Deep visual-semantic quantization for efficient image retrieval," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1328–1337.
[44] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1183–1192.
[45] S. Huang, Y. Xiong, Y. Zhang, and J. Wang, "Unsupervised triplet hashing for fast image retrieval," in *Proc. ACMMM*, 2017, pp. 84–92.
[46] K. Ghasedi Dizaji, F. Zheng, N. Sadoughi, Y. Yang, C. Deng, and H. Huang, "Unsupervised deep generative adversarial hashing network," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 3664–3673.
[47] Y. Zhang and J. R. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 4366–4369.
[48] G. Mantena, S. Achanta, and K. Prahallad, "Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 5, pp. 946–955, 2014.
[49] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Parallel inference of Dirichlet process Gaussian mixture models for unsupervised acoustic modeling: A feasibility study," in *Proc. INTERSPEECH*, 2015, pp. 3189–3193.
[50] Y. Zhang, R. Salakhutdinov, H.-A. Chang, and J. Glass, "Resource configurable spoken query detection using deep Boltzmann machines," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 5161–5164.
[51] H. Wang, T. Lee, and C.-C. Leung, "Unsupervised spoken term detection with acoustic segment model," in *Proc. Int. Conf. Speech Database Assessments*, 2011, pp. 106–111.
[52] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 5157–5160.
[53] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8545–8549.
[54] C.-a. Chan and L.-s. Lee, "Model-based unsupervised spoken term detection with spoken queries," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1330–1342, 2013.
[55] J. Tejedor, M. Fapšo, I. Szöke, J. Černockỳ, F. Grézl *et al.*, "Comparison of methods for language-dependent and language-independent query-by-example spoken term detection," *ACM Trans. Inf. Syst.*, vol. 30, no. 3, p. 18, 2012.
[56] C.-C. Leung *et al.*, "Toward high-performance language-independent query-by-example spoken term detection for mediaeval 2015: Post-evaluation analysis." in *Proc. INTERSPEECH*, 2016, pp. 3703–3707.
[57] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2007, pp. 753–757.
[58] K. Veselỳ, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proc. Autom. Speech Recognit. Understanding*, 2011, pp. 42–47.
[59] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Pairwise learning using multi-lingual bottleneck features for low-resource query-by-example spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5645–5649.
[60] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4950–4954.
[61] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Proc. INTERSPEECH*, 2016, pp. 765–769.
[62] Y.-H. Wang, H.-y. Lee, and L.-s. Lee, "Segmental audio Word2Vec: Representing utterances as sequences of vectors with applications in spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 6269–6273.
[63] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *arXiv preprint arXiv:1703.03130*, 2017.
[64] Y. Zhou, "Clickbait detection in tweets using self-attentive network," *arXiv preprint arXiv:1710.05364*, 2017.
[65] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. INTERSPEECH*, 2018, pp. 3573–3577.
[66] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization (hssas)," *IEEE Access*, vol. 6, pp. 24 205–24 212, 2018.
[67] L. M. Werlen, N. Pappas, D. Ram, and A. Popescu-Belis, "Self-attentive residual decoder for neural machine translation," *arXiv preprint arXiv:1709.04849*, 2017.
[68] J. Wehrmann, M. A. Lopes, M. D. More, and R. C. Barros, "Fast self-attentive multimodal retrieval," in *Proc. Winter Conf. Appl. Comput. Vision*, 2018, pp. 1871–1878.
[69] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Multitask feature learning for low-resource query-by-example spoken term detection," *IEEE J. Sel. Topics Signal Process.*, 2017, vol. 11, no. 8, pp. 1329–1339. .
[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[71] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
[72] Z. Tüske, K. Audhkhasi, and G. Saon, "Advancing sequence-to-sequence based speech recognition," *Proc. INTERSPEECH*, pp. 3780–3784, 2019.

Yougen Yuan received the B.E. degree in Computer Science and Technology from Chongqing University, China, in 2014. He is currently pursuing the Ph.D. degree from the School of Computer Science at Northwestern Polytechnical University, China. In 2016, he visited the Institute for Infocomm Research, Singapore, as an Intern for a year. In 2017, he was a Joint-Training student at the Department of Electrical and Computer Engineering, National University of Singapore. His current research interests include automatic speech recognition and spoken document retrieval.

Lei Xie received the Ph.D. degree in Computer Science from Northwestern Polytechnical University (NPU), Xian, China, in 2004. He is currently a Professor with School of Computer Science, NPU. From 2001 to 2002, he was with the Department of Electronics and Information Processing, Vrije Universiteit Brussel (VUB), Brussels, Belgium, as a Visiting Scientist. From 2004 to 2006, he was a Senior Research Associate in the Center for Media Technology (RCMT), School of Creative Media, City University of Hong Kong, Hong Kong. From 2006 to 2007, he was a Postdoctoral Fellow in the Human Computer Communications Laboratory (HCCL), Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong. He has published more than 200 papers in major journals and proceedings, such as the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, Information Sciences, Pattern Recognition, ACM Multimedia, ACL, INTERSPEECH, and ICASSP. His current research interests include speech and language processing, multimedia and human-computer interaction.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.

Cheung-Chi Leung received the B.E degree from The University of Hong Kong in 1999, and the M.Phil. and Ph.D. degrees from the Chinese University of Hong Kong in 2001 and 2004, respectively. From 2004 to 2008, he was with the Spoken Language Processing group, CNRS-LIMSI, France, as a Postdoctoral Researcher. In 2008, he joined the Institute for Infocomm Research, Singapore, where he worked at the Human Language Technology department for ten years. In 2018, he joined A.I. research lab of Alibaba in Singapore, where he is currently working as a Research Scientist. His current research interests include automatic speech recognition, spoken document retrieval, spoken language recognition.

Hongjie Chen received the B.E degree in the School of Computer Science at Northwestern Polytechnical University, China, in 2013. He is currently a Ph.D. Candidate in the School of Computer Science, Northwestern Polytechnical University. In 2014, he visited the Institute for Infocomm Research, Singapore, as an Intern. In 2017, he served as a Research Student at the Department of Electrical and Computer Engineering, National University of Singapore. His current research interests include automatic speech recognition and spoken document retrieval.

Bin Ma received his Ph.D. degree in computer engineering from The University of Hong Kong, in 2000. He joined Lernout & Hauspie Asia Pacific in 2000 as a Researcher working on speech recognition. From 2001 to 2004, he worked for InfoTalk Corp., Ltd, as a Senior Researcher and a Senior Technical Manager for speech recognition. In 2004, he joined the Institute for Infocomm Research, Singapore, where he worked as a Senior Scientist and Lab Head of Speech Recognition. He has served as Subject Editor for Speech Communication in 2009-2012 and Associate Editor for IEEE/ACM Transactions on Audio, Speech, and Language Processing in 2014-2017. He has also served as Technical Program Co-Chair for INTERSPEECH 2014 and Technical Program Chair for ASRU 2019. He is now working as Principal Engineer at R&D Center Singapore, Machine Intelligence Technology, Alibaba. His current research interests include robust speech recognition, speaker & language recognition, spoken document retrieval, natural language processing, and machine learning.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on July 12,2020 at 01:48:42 UTC from IEEE Xplore. Restrictions apply.