ACOUSTIC SPAN EMBEDDINGS FOR MULTILINGUAL QUERY-BY-EXAMPLE SEARCH

Yushi Hu

University of Chicago, USA

hys98@uchicago.edu

Shane Settle, Karen Livescu

TTI-Chicago, USA

{settle.shane,klivescu}@ttic.edu

arXiv:2011.11807v1 [cs.CL] 24 Nov 2020

ABSTRACT

Query-by-example (QbE) speech search is the task of matching spoken queries to utterances within a search collection. In low- or zero-resource settings, QbE search is often addressed with approaches based on dynamic time warping (DTW). Recent work has found that methods based on acoustic word embeddings (AWEs) can improve both performance and search speed. However, prior work on AWE-based QbE has primarily focused on English data and with single-word queries. In this work, we generalize AWE training to spans of words, producing acoustic span embeddings (ASE), and explore the application of ASE to QbE with arbitrary-length queries in multiple unseen languages. We consider the commonly used setting where we have access to labeled data in other languages (in our case, several low-resource languages) distinct from the unseen test languages. We evaluate our approach on the QUESST 2015 QbE tasks, finding that multilingual ASE-based search is much faster than DTW-based search and outperforms the best previously published results on this task.

***Index Terms*—** query-by-example, acoustic word embeddings, multilingual, low-resource, zero-resource

1. INTRODUCTION

Query-by-example (QbE) speech search is the task of matching spoken queries to utterances in a search collection [1–3]. QbE search often relies on dynamic time warping (DTW) for comparing variable-length audio segments [3–7]. DTW is a dynamic programming approach to determine the similarity between two audio segments by finding their best frame-level alignment [8, 9]. As a result, DTW-based search methods can be very slow (although they can be sped up with approximate nearest neighbor indexing methods [5]), and the results depend on the quality of the frame-level representations [7, 10]. In addition, it can be difficult to modify DTW-based search to cover both exact and approximate query matches well; as a result, the best systems on benchmark QbE tasks often fuse many systems together [7, 11].

An alternative to DTW-based search is to use acoustic word embeddings (AWE) [12–18] — vector representations of spoken word segments — and compare segments using vector distances between their embeddings. This approach to QbE search, first demonstrated using template-based acoustic word embeddings [19] and later using discriminative neural embeddings [20], can greatly speed up search, even over speed-optimized DTW, as well as improve performance. Thus far, this prior work on AWE-based QbE has focused on English data and on single-word queries. Recent work has studied multilingual acoustic word embeddings [21, 22], but thus far has tested them only on proxy word discrimination tasks.

This work makes two main contributions. First, we demonstrate how embedding-based QbE can be effectively applied to multiple unseen languages by using embeddings learned on languages with available labeled data. Second, we extend the idea of acoustic word embeddings to multi-word spans to better handle queries containing an arbitrary numbers of words. We evaluate on the QUESST 2015 benchmark, which includes challenging acoustic conditions, multiple low-resource languages, and both exact and approximate match query settings. Our approach outperforms all prior work on this benchmark, while also being much faster than DTW-based search.

2. RELATED WORK

Low-resource QbE methods often use DTW for audio segment comparison [3, 4, 6, 23]. The frame-level representation is very important in DTW, and a great deal of DTW-based QbE work has focused on learning improved representations. Typical approaches include using phonetic posteriorgrams [3, 4, 24, 25] and supervised or unsupervised bottleneck features (BNF) [7, 10, 11, 26]. In addition, many have found speech activity detection (either energy-based or using a trained phoneme recognizer) to be important for DTW performance and efficiency [7, 11, 27]. Other work has made efficiency improvements through frame-level approximate nearest neighbor search [5]. More challenging query definitions (e.g., approximate matches) and adverse speech conditions (e.g., noisy or reverberant audio) can challenge DTW-based systems and have motivated the use of system ensembles and data augmentation techniques [7, 11, 26]. Symbolic systems based on finite-state transducers built from predicted phone sequences have also been explored to address approximate matches and improve speed, and these can improve performance when fused with more standard DTW systems [7, 27]. As an alternative, the QbE task can be viewed as one of classifying frame-level

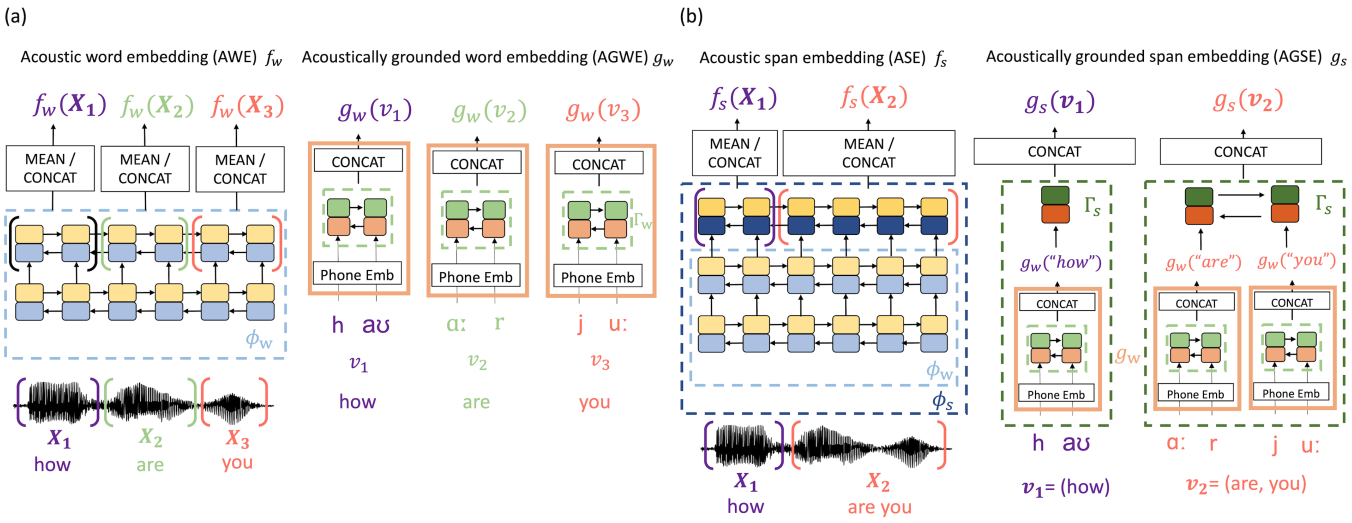

Fig. 1. (a) Contextualized acoustic word embedding (AWE) model f_w and acoustically grounded word embedding (AGWE) model g_w. (b) Acoustic span embedding (ASE) model f_s and acoustically grounded span embedding (AGWE) model g_s.

similarity matrices as matching or non-matching, and learning the classifier in a supervised way [28]; this allows for end-to-end training, but requires task-specific labeled data.

In contrast, embedding-based QbE can be much less complex than methods involving frame-level similarities. Acoustic word embeddings (AWE) have been explored as an alternative to DTW for comparing variable-length acoustic segments. A number of AWE approaches have been developed, primarily for low- and zero-resource settings, including both supervised [13, 15, 29] and unsupervised [12, 16, 18, 30] approaches. AWEs are often evaluated on proxy tasks such as word discrimination [31], but they have also been applied to downstream tasks including QbE [17,19,20] and speech recognition [32,33]. For QbE, AWE-based approaches can improve both search performance and querying efficiency [17,19,20]. Recent work has also explored the use of linguistic knowledge in AWE training to better account for approximate matches [34], although this idea has not yet been applied to QbE search.

Recent work has begun to consider the use of multilingual data for learning AWEs. For example, multilingual BNFs have been used to improve low-resource AWE models on English [17, 35]. Other work has found that AWEs trained on multilingual data can transfer well to additional zero-resource or low-resource languages, when evaluated on word discrimination [21, 22]. However, to our knowledge no prior work has explored AWE-based QbE search in a multilingual setting. In the current work, we extend the multilingual AWEs of [22] in several ways for application to multilingual QbE search.

Several benchmark datasets and tasks have been developed for comparing QbE systems, notably the MediaEval series of challenges [2, 36, 37]. In this work we focus on the QUESST QbE task from MediaEval 2015 [2], which is the most recent QbE task from the MediaEval series and includes a large number of low-resource languages and multiple query settings.

3. MULTILINGUAL EMBEDDING-BASED QBE

Our QbE approach, shown in Figure 2, consists of a multilingual embedding model and a search component. The embedding model maps segments of speech—both the query and segments in the search collection—to vectors, and the search component finds matches between embedding vectors. The embedding model is trained on data from a set of languages, and can then be used for QbE in any language; here we use low-resource languages to train the embedding model, and perform QbE on a disjoint set of unseen languages. The overall approach is very simple, using no speech activity detection, fine-tuning for the QbE task, or special-purpose components to accommodate approximate matches; we simply use the trained embeddings out of the box, plugging them into the parameter-free search component.

The following subsections describe each component. For the embedding model, we begin with acoustic word embeddings as has been done in prior work on embedding-based QbE [17, 19, 20]. We then extend the approach to accommodate spans of multiple words, resulting in *acoustic span embeddings*. Finally, we describe the search component.

3.1. Acoustic word embedding (AWE) model

For the acoustic word embedding model, we take as our starting point an approach that produces the best word discrimination results in prior work, based on jointly learning AWEs and written word embeddings (referred to as acoustically grounded word embeddings (AGWEs)) using a multi-view contrastive loss [15, 22], where the AWE is the acoustic "view" and the AGWE is the written "view". We note that we do not use the written embedding model, but we jointly train both models

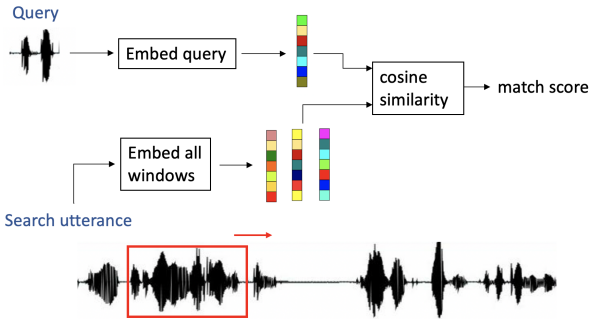

Fig. 2. QbE search with acoustic embeddings.

since this approach has produced high-quality AWEs in prior work. We slightly extend the approach in order to embed word segments in context. Contextual AWEs improve QbE performance [17], simplify extension to multi-word spans (Section 3.2), and help efficiently embed the search collection.

For AWE learning, we are given a training set of N spoken utterances, $\{\mathbf{X}^{(n)}\}_{n=1}^{N}$, where each utterance $\mathbf{X} \in \mathbb{R}^{T \times D}$ has a corresponding word-level alignment $\mathcal{A}$ used to extract spoken word segments. A word alignment $\mathcal{A} = \{(s_1, e_1, v_1), ..., (s_L, e_L, v_L)\}$ consists of tuples (s_i, e_i, v_i) indicating start frame, end frame, and word label, respectively, for each word in the utterance.

Figure 1 outlines the structure of the embedding models. The acoustic-view model f_w consists of an utterance encoder Φ_w and a pooling function G. The embedding of the i^{th} segment in $\mathbf{X}$ is given by the pooled encoder outputs over the frames in the segment:

$$f(\mathbf{X}_i) = f(\mathbf{X}, \mathcal{A}_i) = G(\Phi_w(\mathbf{X}), s_i, e_i)$$

where $\Phi_w(\mathbf{X}) \in \mathbb{R}^{T \times d}$, $\mathcal{A}_i = (s_i, e_i, v_i)$, and G is applied over the window $[s_i, e_i]$. For the encoder Φ_w, we use a bidirectional recurrent network, so output features within $[s_i, e_i]$ depend on the full utterance context. We consider two pooling functions G, either concatenation of the starting and ending encoder states or a mean over all encoder states in the segment:

$$G(\Phi_w(\mathbf{X}), s_i, e_i) = \left[\Phi_w(\mathbf{X})_{e_i}^{\rightarrow}; \Phi_w(\mathbf{X})_{s_i}^{\leftarrow}\right] \quad \text{(concat)}$$

$$G(\Phi_w(\mathbf{X}), s_i, e_i) = \frac{1}{e_i - s_i + 1} \sum_{j=s_i}^{e_i} \Phi_w(\mathbf{X})_j \quad \text{(mean)}$$

where $[\mathbf{u}; \mathbf{v}]$ indicates concatenation of vectors $\mathbf{u}$ and $\mathbf{v}$ and, for bidirectional network outputs, $\rightarrow$ and $\leftarrow$ index the forward and backward hidden states, respectively.

The written-view model g_w consists of a pronunciation lexicon $\text{Lex}(\cdot)$ mapping words to phones[1], an encoder Γ_w, and a pooling function (concatenation). As in prior work [15, 22], the embedding of the word v_i is defined as

$$g(v_i) = [\Gamma_w(\mathbf{p}_i)_l^{\rightarrow}; \Gamma_w(\mathbf{p}_i)_1^{\leftarrow}]$$

where $\mathbf{p}_i = \text{Lex}(v_i)$, $l = \text{len}(\mathbf{p}_i)$, and $\Gamma_w(\mathbf{p}_i) \in \mathbb{R}^{l \times d}$.

During training, we minimize the following objective consisting of three contrastive loss terms (updated slightly from [22] to improve word discrimination performance):

$$\sum_{n=1}^{N} \sum_{i=1}^{|\mathcal{A}|} \sum_{obj=0}^{2} \mathcal{L}_{obj}(\mathbf{X}_i^{(n)}, v_i^{(n)}) \quad (1)$$

Semi-hard negative sampling [38] is done for all three terms:

$$\mathcal{L}_0(\mathbf{X}, v) = \left[m + d(f(\mathbf{X}), g(v)) - \min_{v' \in \mathcal{V}_0'(\mathbf{X}, v)} d(f(\mathbf{X}), g(v'))\right]_+$$

$$\mathcal{L}_1(\mathbf{X}, v) = \left[m + d(f(\mathbf{X}), g(v)) - \min_{v' \in \mathcal{V}_1'(\mathbf{X}, v)} d(g(v), g(v'))\right]_+$$

$$\mathcal{L}_2(\mathbf{X}, v) = \left[m + d(f(\mathbf{X}), g(v)) - \min_{\mathbf{X}' \in \mathcal{X}_2'(\mathbf{X}, v)} d(g(v), f(\mathbf{X}'))\right]_+$$

where f is f_w, g is g_w, m is a margin hyperparameter, d denotes cosine distance $d(a, b) = 1 - \frac{a \cdot b}{\|a\| \|b\|}$, and

$$\mathcal{V}_0'(\mathbf{X}, v) := \{v' | d(f(\mathbf{X}), g(v')) > d(f(\mathbf{X}), g(v)), v' \in \mathcal{V}/v\}$$
$$\mathcal{V}_1'(\mathbf{X}, v) := \{v' | d(g(v), g(v')) > d(g(v), f(\mathbf{X})), v' \in \mathcal{V}/v\}$$
$$\mathcal{X}_2'(\mathbf{X}, v) := \{\mathbf{X}' | d(g(v), f(\mathbf{X}')) > d(g(v), f(\mathbf{X})), v' \in \mathcal{V}/v\}$$

where N is the number of utterances and $\mathcal{V}$ is the training vocabulary. In practice, this objective is applied over a mini-batch such that N represents that batch size and $\mathcal{V}$ represents the unique words in that batch. Also, instead of using the single most offending semi-hard negative, we find the top k within the batch and each contrastive loss term in Equation 1 is an average over these k negatives. Each mini-batch is from a single language, similar to [22].

This objective encourages embedding spoken word segments corresponding to the same word label close together and close to their learned label embeddings, while ensuring that segments corresponding to different word labels are mapped farther apart (and nearer to their respective label embeddings).

3.2. Acoustic span embedding (ASE) model

Next, we extend the idea of jointly trained AWE+AGWE to jointly trained acoustic span embeddings (ASE) and acoustically grounded span embeddings (AGSE) to better model spans of multiple words that may appear in queries and search utterances. As before, we train both embedding models jointly, but use only the ASE model for QbE.

The training data for ASE+AGSE learning consists of spans of consecutive word segments, constructed from the word-aligned AWE+AGWE training data by merging neighboring word segments. Specifically, given the training utterance $\mathbf{X}$ with word-level alignment $\mathcal{A}$, we merge any given pair of consecutive word segments i and j with probability p, to generate a span-level alignment $\mathcal{A}'$

$$\mathcal{A}' = \{\ldots, (s_i, e_j, \mathbf{v}_{i:j}), \ldots\} = \{\ldots, (s_{i'}, e_{i'}, \mathbf{v}_{i'}), \ldots\}$$

[1]The approach can be relaxed to not require a pronunciation lexicon, using instead the written character sequence. Prior work has found an advantage to phones [22], but the performance difference is not large.

where $|\mathcal{A}'| \leq |\mathcal{A}|$ and $\mathcal{A}'$ consists of tuples $(s_{i'}, e_{i'}, \mathbf{v}_{i'})$ indicating start frame, end frame, and multi-word label sequence, respectively, for each span in the utterance after merging.

As shown in Figure 1, the ASE model f_s has the same form as the AWE function, consisting of an encoder Φ_s and a pooling function G:

$$f(\mathbf{X}_{i'}) = f(\mathbf{X}, \mathcal{A}_{i'}) = G(\Phi_s(\mathbf{X}), s_{i'}, e_{i'})$$

where $\Phi_s(\mathbf{X}) \in \mathbb{R}^{T \times d}$, $\mathcal{A}'_{i'} = (s_{i'}, e_{i'}, \mathbf{v}_{i'})$, and G is applied over the window $[s_{i'}, e_{i'}]$.

The written-view model (the acoustically grounded span embedding model) g_s consists of an encoder Γ_s and another pooling function (which is, again, concatenation):

$$g_s(\mathbf{v}_{i'}) = [\Gamma_s(\mathbf{v}_{i'})^{\rightarrow}_{l'}; \Gamma_s(\mathbf{v}_{i'})^{\leftarrow}_{1}]$$

where $\mathbf{v}_{i'} = (v_i, \ldots, v_j), l' = j - i + 1$, and $\Gamma(\mathbf{w}_{i'}) \in \mathbb{R}^{l' \times d}$.

We jointly train f_s and g_s by minimizing a span-level contrastive loss, which has the same form as Equation 1:

$$\sum_{n=1}^{N} \sum_{i'=1}^{|\mathcal{A}'|} \sum_{obj=0}^{2} \mathcal{L}_{obj}(\mathbf{X}_{i'}^{(n)}, \mathbf{v}_{i'}^{(n)}) \tag{2}$$

where the loss terms are defined as in Section 3.1 but where g is g_s, f is f_s, and $\mathcal{V}$ is the set of multi-word spans $\mathbf{v}$. ASE+AGSEs can be trained from scratch using this objective, but in this work we first pre-train word-level models f_w, g_w and use them to initialize the lower layers of Φ_s, Γ_s. One potential pitfall of a span-level contrastive loss would seem to be that negative examples may be "too easy". In preliminary experiments, we explored alternative objective functions taking into account not only same and different span pairs but also the degree of difference between them, but these did not outperform the simple contrastive loss described here.

3.3. Embedding-based QbE

Figure 2 depicts our multilingual embedding-based QbE system. [2] Given a pre-trained embedding model (either AWE or ASE), we first build an index of utterances in the search collection by embedding all possible segments that meet certain minimal constraints. Similarly to [17], we use a simple sliding window algorithm with several window sizes to get possible matching segments. The segment in each analysis window is then mapped to an embedding vector using one of our AWE or ASE embedding models. Contextual AWEs and ASEs help to simplify this process, since each utterance can be forwarded just once through the embedding network (ϕ_w or ϕ_s); the segment embedding for each segment in the utterance is then computed by pooling over the relevant windows.

At query time, given an audio query, we embed the query with the same pre-trained embedding model, and then compute a detection score for each utterance in the search collection via cosine similarity between the corresponding embeddings:

$$score(\mathbf{q}, \mathbf{S}) = \max_{\mathbf{s} \in \Sigma_{\mathbf{q}}} \frac{f(\mathbf{s}) \cdot f(\mathbf{q})}{\|f(\mathbf{s})\|_2 \|f(\mathbf{q})\|_2} \tag{3}$$

where f is the acoustic-view embedding model (either AWE or ASE), $\mathbf{q}$ is the spoken query, $\mathbf{S}$ is the search utterance, and $\Sigma_{\mathbf{q}}$ is the set of all windowed segments in S of similar length to $\mathbf{q}$. We compute this score for all possible segments in the search collection, but it is possible to speed up the search via approximate nearest neighbor search (as in previous work [19, 20]). In this work, we are not concerned with obtaining the fastest QbE system possible, so we simply use exhaustive search; but as we will see in Section 5.4, this exhaustive embedding-based search is still much faster than DTW-based alternatives.

4. EXPERIMENTAL SETUP

4.1. Embedding models

We train our embedding models (Figure 1) on conversational speech from 12 languages including 22 hours of English from Switchboard [39] and 20-120 hours from each of 11 languages[3] in the IARPA Babel project [40]. We use Kaldi Babel recipe `s5d` [41] to compute acoustic features and extract word alignments for the Babel languages. We use 39-dimensional acoustic features consisting of 36-dimensional log-Mel spectra and 3-dimensional (Kaldi default) pitch features. All other details are the same as [22]. In some experiments, we augment the data with SpecAugment [42], using one frequency mask ($m_F = 1$) with width $F \sim \mathcal{U}\{0, 9\}$ and one time mask ($m_T = 1$) with width $T \sim \mathcal{U}\{0, \frac{t_{min}}{2}\}$, where t_{min} is the length of the shortest word segment in the utterance such that no word is completely masked.

For AWE+AGWE training, the acoustic-view model (f_w) encoder Φ_w is a 4-layer bidirectional gated recurrent unit (BiGRU [43]) network with dropout rate 0.4, and the pooling function G is either mean or concatenation. The written-view model (g_w) encoder Γ_w is a phone embedding layer followed by a 1-layer BiGRU. For ASE+AGSE training, the acoustic-view model (f_s) encoder Φ_s is a 6-layer BiGRU network with dropout rate 0.4. The written-view model (g_s) encoder Γ_s is a 1-layer BiGRU on top of a AGWE submodule that has the same structure as g_w. Before span-level training, we first train AWE+AGWE models f_w and g_w to be used as initialization. The bottom 4 layers of Φ_s are initialized with Φ_w (from f_w) and are kept fixed during training. The bottom word-level submodule in Γ_s is also initialized with g_w and not updated during training. All recurrent models use 256 hidden dimensions per direction per layer and generate 512-dimensional embeddings.

During ASE+AGSE training, to convert word-level alignment $\mathcal{A}$ (length L) into span-level alignment $\mathcal{A}'$, we construct the list of word boundaries $\mathcal{B}_{\mathcal{A}} = \{(e_1, s_2), \ldots, (e_{L-1}, s_L)\}$. Next, we sample the number of word boundaries to remove $r \sim \mathcal{U}(\frac{L-1}{2}, L-1)$, and then sample r tuples from $\mathcal{B}_{\mathcal{A}}$. For

[2] `https://github.com/Yushi-Hu/Query-by-Example`

[3] Cantonese (122 hrs), Assamese (50 hrs), Bengali (51 hrs), Pashto (68 hrs), Turkish (68 hrs), Tagalog (74 hrs), Tamil (56 hrs), Zulu (52 hrs), Lithuanian (33 hrs), Guarani (34 hrs), and Igbo (33 hrs).

each of the tuples (e_i, s_{i+1}), we merge the segments i and $i+1$ such that $\{\ldots,(s_i,e_i,v_i),(s_{i+1},e_{i+1},v_{i+1}),\ldots\}$ becomes $\{\ldots,(s_i,e_{i+1},\mathbf{v}_{i:i+1}),\ldots\}$, giving us $\mathcal{A}'$. The margin m in Equation 1 is 0.4 and the mini-batch size is ~ 30000 acoustic frames. We start with sampling $k = 64$ negatives and gradually reduce to $k = 20$. We use Adam [44] to perform mini-batch optimization, with initial learning rate 0.0005 and $L2$ weight decay parameter 0.0001. All other settings are the same as in [22]. Hyperparameters are tuned using cross-view word discrimination performance on the dev sets as in [22].

4.2. QbE system

We preprocess each utterance in the search collection, using a sliding window approach, to generate a set of possible segments and embed all of the segments using one of our (AWE/ASE) embedding models. Specifically, we consider the set of possible segments to be windows of size $\{12, 15, 18, \ldots, 30, 36, 42, 48, \ldots, 120\}$ frames, with shifts of 5 frames between windows. We embed the query (length l_q) using the same AWE/ASE model and compute the cosine similarity between the query embedding and those of all windowed segments with length between $\frac{2}{3}l_q$ and $\frac{4}{3}l_q$.

4.3. Evaluation

We evaluate our models on the Query by Example Search on Speech Task (QUESST) at Mediaeval 2015 [2]. The task uses a dataset containing speech in 6 languages (Albanian, Czech, Mandarin, Portuguese, Romanian, and Slovak). There are about 18 hours of test utterances in the search collection, 445 development queries, and 447 evaluation queries. The dataset includes artificially added noise and reverberation with equal amounts of clean, noisy, reverberated, and noisy+reverberated speech. The task includes three types of sub-tasks: **T1** (exact match), **T2** (allowing word reordering and lexical variations), and **T3** (like **T2**, but conversational queries in context).

4.3.1. Evaluation metrics

We use the official QUESST 2015 grading scripts [2]. The QbE system outputs a score (in our case, a cosine similarity) for each pair of query $\mathbf{q}$ and utterance $\mathbf{S}$. A threshold θ can be set such that if $score(\mathbf{q}, \mathbf{S}) > \theta$, then q is considered a hit. The evaluation script varies θ to compute normalized cross entropy (C_{nxe}) and term-weighted value (TWV) as defined in [2, 36]. In QUESST 2015, C_{nxe} is considered the primary metric. The final results are reported as $minC_{nxe} = \min_\theta C_{nxe}$ and $maxTWV = \max_\theta TWV$, that is the best values achieved for the metrics by varying the threshold.

Specifically, the normalized cross entropy C_{nxe} is the ratio between the cross entropy of the QbE system output scores (measured against the binary ground truth) and the cross entropy of random scoring. It ranges between $[0, 1]$, and lower C_{nxe} values are better. The term weighted value TWV is based on hard decisions and is defined as $1 - (P_{miss}(\theta) + \beta P_{fa}(\theta))$ where θ is a threshold, P_{miss} is the miss rate, P_{fa} is the false alarm rate, and β is a term weighing the cost of misses vs. false alarms. TMV ranges from $-\beta$ to 1, and higher values are better. The official script uses $\beta = 12.49$.

5. RESULTS

5.1. Comparison with prior work

Table 1 compares our work with the top QUESST 2015 submissions as well as the best follow-up work [7]. Both the AWE and ASE models we use here are 6 layers for a fair comparison. Many QUESST systems are trained on labelled data (both in- and out-of-domain with respect to the target languages) and use augmentation techniques to limit mismatch with the test data. In addition, speech activity detection (SAD) and system fusion significantly improve performance, and all of the top-performing QUESST systems use both.

Our AWE (mean) model with SpecAugment is competitive with some of the top prior $minC_{nxe}$ results, but there is a clear benefit to span-level embeddings. Although our approach is much simpler, our best single system, ASE (mean) with SpecAugment, outperforms all prior work on the primary metric $minC_{nxe}$. On the $maxTWV$ metric, this system matches all but two of the fusion models in prior work ([26] and [7]). To compete with these fusion systems, we create our own simple fusion model by summing the scores output by our two ASE (mean) and ASE (concat) systems (both trained with SpecAugment), which gives us an additional performance boost, matching $maxTWV$ of the best prior fusion models and outperforming them in $minC_{nxe}$ by a wide margin.

5.2. Dependence on query sub-task

Figure 3 compares $minC_{nxe}$ performance across the QUESST sub-tasks, for both our best single model, ASE (mean) with SpecAugment, and the best models from [7]. We can see that the exact DTW system does best in the exact match sub-task (**T1**) but cannot generalize well to approximate matches (**T2** and **T3**), while the partial-match system has the opposite behavior. This illustrates the importance of model fusion in DTW-based systems as each individual system can specialize to a particular sub-task. Meanwhile, our best single system is competitive with the others on exact match (**T1**) and outperforms all of them on approximate match tasks. This suggests that ASE models are better at accommodating lexical variations and word re-ordering than DTW-based systems without any special-purpose partial-match handling, and without sacrificing too much performance on exact matches.

5.3. Contribution of segment embeddings

Our approach differs from prior work in a number of respects, including choice of training data and newer neural network techniques. One may ask how much of the performance improvements are due to these differences, rather than due to our segment embedding-based approach. To address this question, we perform a simple comparison on the proxy task of word discrimination, which is often used to compare speech representations [13, 31]. The task is, given a pair of speech

Table 1. QUESST 2015 performance on dev and eval sets measured by $minC_{nxe}$ and $maxTWV$. Training languages are separated into in- and out-of-domain. All SAD systems are based on phone recognizers.

Method	**systems**	**languages** in	out	**labeled data**[4] hours	**SAD**	**Augmentation**	$minC_{nxe} \downarrow$ / $maxTWV \uparrow$ dev	eval
Top prior results								
BNF+DTW [11]	36	2	4	384+	Yes	noise	0.778 / 0.234	0.787 / 0.206
BNF+DTW [26]	66	2	15	643+	Yes	noise + reverb	0.757 / 0.286	0.747 / 0.274
Exact match fusion [7]	2	0	2	423	Yes	noise + reverb	0.795 / 0.256	
Partial match + symbolic [7]	2	0	1	260	Yes	noise + reverb	0.783 / 0.231	
Fusion of above two [7]	4	0	2	423	Yes	noise + reverb	0.723 / 0.320	
Our systems								
AWE (concat)	1	0	12	664			0.845 / 0.084	
AWE (mean)	1	0	12	664			0.803 / 0.101	
AWE (mean)	1	0	12	664		SpecAugment	0.782 / 0.135	
ASE (concat)	1	0	12	664			0.753 / 0.193	
ASE (mean)	1	0	12	664			0.728 / 0.239	
ASE (mean)	1	0	12	664		SpecAugment	**0.706 / 0.255**	**0.692 / 0.246**
ASE (mean+concat)	2	0	12	664		SpecAugment	**0.670 / 0.323**	**0.658 / 0.298**

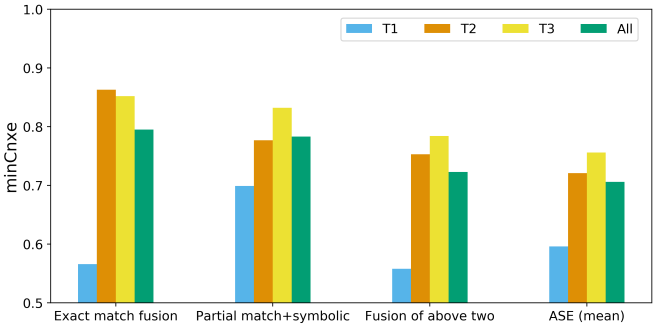

Fig. 3. Performance on the three QUESST 2015 sub-tasks, measured by dev set $minC_{nxe}$ (↓).

segments each corresponding to a word, to determine whether they correspond to the same word or not. Specifically we use the task and multilingual development set of [22]. We compare performance on this task achieved using a vector cosine distance between our acoustic word embeddings, versus using DTW applied to the final layer hidden states of our network. Both approaches use the same network trained on the same data, so the only difference is whether we measure distances using segment embeddings computed from the network or DTW on frame-level network outputs. On this task, the acoustic word embedding approach has an average precision (AP) of 0.57, whereas the DTW approach has an AP of 0.49. This comparison demonstrates the improvement due to embedding segments as vectors rather than aligning them via DTW.

5.4. Run time

Table 2 compares the average per-query run times of our implementations of ASE-based and DTW-based QbE search. In the DTW-based system, we use a sliding window approach with a window size of 90 frames and a shift of 10 frames. This results in each query being compared with $600K$ windowed segments, while our ASE-based QbE system on average compares each query with $4M$ windowed segments as described in Section 4.2. We use 39-dimensional filterbank+pitch features and 512-dimensional BiGRU hidden states from the ASE model as inputs to the DTW-based QbE system. All systems are tested on a single thread of an Intel Core i7-9700K CPU. As expected, the ASE-based approach is more than two orders of magnitude faster than the DTW-based approaches. Although the DTW system here is not identical to any of the actual QUESST 2015 systems, this rough comparison gives an idea of the speed difference. In addition, for emedding-based systems used in a real search setting, approximate nearest neighbor can be used to further speed up the search [19, 20].

Table 2. Run times on the QUESST 2015 development set.

Method	# of comparisons (per query)	Run-time (s / query)
DTW on Filterbank features	600K	486
DTW on ASE hidden states	600K	847
ASE-based QbE	4M	**5**

6. CONCLUSION

We have presented a simple embedding-based approach for multilingual query-by-example search that outperforms prior work on the QUESST 2015 QbE task, while also being much more efficient. We verify that multilingual acoustic word embedding (AWE) models can be effective for query-by-example search on unseen target languages, and we further improve performance by extending the approach to multi-word spans using acoustic span embeddings (ASE). Our embeddings are jointly trained with embeddings of the corresponding written words, but for QbE we ignore the written embeddings. In future work it will be interesting to use both of the jointly learned models to search by either spoken or written query.

[4] "+" denotes that some dataset sizes are unavailable.

7. REFERENCES

[1] Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran, "Query-by-example spoken term detection for oov terms," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009.

[2] Xavier Anguera Miró, Luis Javier Rodriguez-Fuentes, Igor Szöke, Andi Buzo, and Florian Metze, "Query by example search on speech at MediaEval 2015," in *MediaEval*, 2015.

[3] Timothy J Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009.

[4] Yaodong Zhang and James R Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009.

[5] Aren Jansen and Benjamin Van Durme, "Indexing raw acoustic features for scalable zero resource search," in *Proc. Interspeech*, 2012.

[6] Gautam Mantena and Xavier Anguera, "Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[7] Cheung-Chi Leung, Lei Wang, Haihua Xu, Jingyong Hou, Van Tung Pham, Hang Lv, Lei Xie, Xiong Xiao, Chongjia Ni, Bin Ma, et al., "Toward high-performance language-independent query-by-example spoken term detection for MediaEval 2015: Post-evaluation analysis.," in *Proc. Interspeech*, 2016.

[8] Taras K Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics*, vol. 4, no. 1, pp. 52–57, 1968.

[9] Hiroaki Sakoe and Seibi Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[10] Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Unsupervised bottleneck features for low-resource query-by-example spoken term detection.," in *Proc. Interspeech*, 2016.

[11] Jorge Proença and Fernando Perdigão, "Segmented dynamic time warping for spoken query-by-example search.," in *Proc. Interspeech*, 2016.

[12] Keith Levin, Katharine Henry, Aren Jansen, and Karen Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013.

[13] Herman Kamper, Weiran Wang, and Karen Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[14] Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung yi Lee, and Lin-Shane Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Interspeech*, 2016.

[15] Wanjia He, Weiran Wang, and Karen Livescu, "Multi-view recurrent neural acoustic word embeddings," in *"Proc. Int. Conf. on Learning Representations (ICLR)"*, 2017.

[16] Herman Kamper, "Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[17] Yougen Yuan, Cheung-Chi Leung, Lei Xie, Hongjie Chen, Bin Ma, and Haizhou Li, "Learning acoustic word embeddings with temporal context for query-by-example speech search," in *Proc. Interspeech*, 2018.

[18] Nils Holzenberger, Mingxing Du, Julien Karadayi, Rachid Riad, and Emmanuel Dupoux, "Learning word embeddings: Unsupervised methods for fixed-size representations of variable-length speech segments," in *Proc. Interspeech*, 2018.

[19] Keith Levin, Aren Jansen, and Benjamin Van Durme, "Segmental acoustic indexing for zero resource keyword search," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[20] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proc. Interspeech*, 2017.

[21] Herman Kamper, Yevgen Matusevych, and Sharon Goldwater, "Multilingual acoustic word embedding models for processing zero-resource languages," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[22] Yushi Hu, Shane Settle, and Karen Livescu, "Multilingual jointly trained acoustic and written word embeddings," in *Proc. Interspeech*, 2020.

[23] Yaodong Zhang and James Glass, "A piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping," in *Proc. Interspeech*, 2011.

[24] Marcos Calvo Lafarga, Mayte Giménez, Lluís F. Hurtado, Emilio Sanchis Arnal, and Jon Ander Gómez, "ELiRF at MediaEval 2015: Query by example search on speech task (QUESST)," in *MediaEval*, 2015.

[25] Paula Lopez-Otero, Laura Docio-Fernandez, and Carmen García-Mateo, "GTM-UVigo systems for the query-by-example search on speech task at MediaEval 2015," in *MediaEval*, 2015.

[26] Jingyong Hou et al., "The NNI query-by-example system for MediaEval 2015," in *Working Notes Proceedings of the MediaEval 2015 Workshop*, 2015.

[27] Haikua Xu et al., "Approximate search of audio queries by using DTW with phone time boundary and data augmentation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[28] Dhananjay Ram, Lesly Miculicich, and Hervé Bourlard, "Neural network based end-to-end query by example spoken term detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1416–1427, 2020.

[29] Shane Settle and Karen Livescu, "Discriminative acoustic word embeddings: Recurrent neural network-based approaches," in *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, 2016.

[30] Kartik Audhkhasi, Andrew Rosenberg, Abhinav Sethy, Bhuvana Ramabhadran, and Brian Kingsbury, "End-to-end ASR-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[31] Michael A Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. Interspeech*, 2011.

[32] Samy Bengio and Georg Heigold, "Word embeddings for speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[33] Shane Settle, Kartik Audhkhasi, Karen Livescu, and Michael Picheny, "Acoustically grounded word embeddings for improved acoustics-to-word speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[34] Zixiaofan Yang and Julia Hirschberg, "Linguistically-informed training of acoustic word embeddings for low-resource languages," in *Proc. Interspeech*, 2019.

[35] Yougen Yuan, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Learning neural network representations using cross-lingual bottleneck features with word-pair information," in *Proc. Interspeech*, 2016.

[36] Luis J Rodriguez-Fuentes and Mikel Penagarikano, "MediaEval 2013 spoken web search task: system performance measures," *n. TR-2013-1, Department of Electricity and Electronics, University of the Basque Country*, 2013.

[37] Xavier Anguera, Luis Javier Rodriguez-Fuentes, Igor Szöke, Andi Buzo, and Florian Metze, "Query by example search on speech at MediaEval 2014," in *MediaEval*, 2014.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[39] John J Godfrey, Edward C Holliman, and Jane McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 1992.

[40] "IARPA Babel language pack: IARPA-babel101b-v0.4c, IARPA-babel102b-v0.5a, IARPA-babel103b-v0.4b, IARPA-babel104b-v0.4by, IARPA-babel105b-v0.5, IARPA-babel106-v0.2g, IARPA-babel204b-v1.1b, IARPA-babel206b-v0.1e, IARPA-babel304b-v1.0b, IARPA-babel305b-v1.0c, IARPA-babel306b-v2.0c," .

[41] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[42] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019.

[43] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[44] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *"Proc. Int. Conf. on Learning Representations (ICLR)"*, 2015.