Enc-Dec RNN Acoustic Word Embeddings learned via Pairwise Prediction

Adhiraj Banerjee, Vipul Arora

Department of Electrical Engineering
Indian Institute of Technology Kanpur, India
{adhiraj,vipular}@iitk.ac.in

Abstract

Learning discriminative Acoustic Word Embeddings (AWEs) summarising variable length spoken word segments brings efficiency in speech retrieval tasks, notably, Query-by-Example (QbE) Speech or Spoken Term Detection (STD). In this paper, we add on to RNN based approaches for generating acoustic word embeddings. The model is trained in an encoder-decoder fashion on pairs of similar word segments by optimizing a pairwise self-supervised loss where the targets are generated offline via clustering. The pairs may be generated with word boundaries (weak supervision) or via augmentation of unlabelled word segments (no supervision). Experiments with word discrimination task on TIMIT and LibriSpeech show state of the art performance of the proposed approach outperforming popular RNN AWE approaches in both weakly supervised and unsupervised settings. The AWEs generated by our model generalise well to OOV words. On STD tasks performed on TIMIT, the proposed approach provides speed advantages.
Index Terms: word embeddings, self-supervised learning, spoken term detection, query-by-example, pairwise learning

1. Introduction

Spoken Term Detection (STD) is defined as the task of searching for a spoken query term (a word say, "cat") in a collection of long speech utterances (say, recordings of hour long speech). It is a zero resource task with the query term and search space given as audio data with no language specific resources. Unlike Keyword Spotting, the spoken query terms may also be out-of-vocabulary (OOV) words [1, 2].

Most approaches to STD typically involve two steps: (a) extraction of sequences of feature vectors from query and speech utterance, and (b) detection of query in the speech utterance using different variations of Dynamic Time Warping (DTW) based template matching techniques. The features extracted may be spectral features [3, 4], posterior features (vectors indicating posterior probabilities of sub-word units like phonemes) [5, 6] as well as bottleneck features [7]. The posterior features may be extracted using Gaussian Mixture Models (GMMs) [5] requiring no supervision or Deep Neural Networks (DNNs), either trained with supervision [6, 8] or trained in an unsupervised fashion via spectral clustering [9]. After extraction of sequences of feature vectors from query and speech utterance, a frame-level similarity matrix is generated with the pair of sequences over which variations of DTW such as Segmental DTW [5], [3], Slope-constrained DTW [10], and Sub-sequence DTW (S-DTW)[11] are used to detect the occurrence of query.

A faster alternative to DTW based approaches is learning of fixed dimensional vector representations for variable length spoken word segments i.e. Acoustic Word Embeddings (AWEs). Representing word segments with their AWE brings drop in search time as the similarity between two segments boils down to calculating a cosine or euclidean distance. It also allows us to consider a flexible set of features and make decision on the basis of information encompassing longer spans of time.

Neural Acoustic Word Embedding (NAWE) models are a set of encoder models f which input a variable length sequence of acoustic features $X = [x_t; t \in [T]]$ of length T time steps (word utterances) and output a single fixed-dimensional embedding (AWE) $f(X) = e \in R^d$ summarising the acoustic feature sequence X. Works like [12, 13, 14, 15, 16, 17, 18, 19, 20] present approaches to learning discriminative AWEs from data using a number of neural embedding models and training tasks. In this paper, we add on to Recurrent Neural Network (RNN) based approaches to generating AWEs and compare our approach with state of the art approaches [17, 18, 19, 20]. RNNs are natural candidates for generation of AWEs as by nature they can handle arbitrary length input sequences.

We use an encoder-decoder RNN architecture as our NAWE model where the last time-step encoder representation generated given input word segment is used as AWE. Our model is trained on pairs of same word utterances which differ in pitch and rate. We use a novel pairwise prediction task inspired from pseudo-labelling based self-supervised speech representation learning approaches like HuBERT [21]. In this task , given a pair of same word utterances, we predict a pseudo-label assigned to each time-step t in each sequence with t time-step decoder representation generated from the other sequence in the pair. The pseudo-labelling is performed with a K-means clustering model trained offline over MFCCs.

We conduct experiments on Word Discrimination task [22] over OOV words extracted from LibriSpeech and TIMIT datasets to present how discriminatory and generalizable the AWEs learnt via our approach is in comparison to SOTA baselines. Our training approach outperforms [17, 18, 19, 20] and MFCCs in terms of Average Precision (AP). We conduct STD on TIMIT dataset to present speed advantages in using AWEs.

2. Embedding-based STD

Our STD system needs to quickly retrieve from a large collection those segments nearest to a given spoken query. All the segments and spoken query are converted to AWEs. The embedding based query-by-example approach, Segmental Randomized Acoustic Indexing and Logarithmic-Time Search (S-RAILS) system [23] is used for retrieval. S-RAILS uses a version of locality-sensitive hashing (LSH) [24, 25] to perform approximate nearest neighbor search over the AWEs.

The long utterance where we are to detect the query term is represented as a large search collection of AWEs $X = \{x_i \in$

$R^d; i \in [N]\}$. The spoken query utterance is also converted to an AWE $q \in R^d$. S-RAILS uses LSH to provide a fast approximation of the computationally expensive d-dimensional cosine distance between AWEs. All the AWEs in R^d are converted to bit vectors or signatures using LSH, such that if $x_i, x_j \in X$ are close under the cosine distance, their signatures $s_i, s_j \in \{0,1\}^b$ will agree in most of their entries.

S-RAILS arranges the signatures $s(X) = \{s_i; i \in [N]\}$ into a lexicographically sorted list S. The query AWE q is then converted to its LSH signature $s_q \in \{0,1\}^b$ and its location in the sorted list S is retrieved in $O(\log b)$ time. A set of approximate nearest neighbors can be retrieved by looking at B entries in the list S before s_q and B entries after s_q. Note, bits appearing earlier in the signatures have far more influence on whether $x_i, x_j \in X$ will be judged similar. S-RAILS performs the lexicographic lookup under P different permutations of the bits to eradicate this effect.

S-RAILS has 3 parameters: the signature length b, search beam-width B and the number of permutations of the bits P. Increasing any of these parameters improves the approximation to the cosine distance with a trade-off in memory required and run-time. Building the index requires at-most $O(PbN \log N)$ time while querying the index requires $O(B + Pb \log N)$ time.

3. Related Works and Baselines

RNN based NAWE models use a BiLSTM encoder network which takes in MFCC feature sequence X and outputs context representation $z_t \in R^d$ at each time step t. This results in an output sequence of representations $Z = [z_t; t \in [T]]$. The last time step representation $z_T \in Z$ is chosen as the acoustic word embedding e corresponding to the sequence X. The model is represented as,

$$f(X) = e = z_T \tag{1}$$

[17, 18] are two weakly-supervised state of the art approaches in training such NAWE models. Both approaches require word boundaries and word label information as the models are trained on batches of triplets (a, p, n) where anchor a is a word utterance, positive p is an utterance of the same word as anchor, and negative n is an utterance of another word. To form the triplets we further require the word label of each word utterance in dataset.

[17] trains their model f with a siamese weight sharing scheme on triplets of MFCC feature sequences (a, p, n) where (a, p) is a positive pair corresponding to the same word and n is a negative corresponding to another word. A cosine hinge loss function is optimized to learn discriminatory AWEs which is given as follows,

$$\mathcal{L}_{hinge} = \max\{0, m + d_{cos}(f(a), f(p)) - d_{cos}(f(a), f(n))\} \tag{2}$$

where $d_{cos}(j,k) = 1 - cos(j,k)$, $cos(j,k) = \frac{j \cdot k}{|j||k|}$ is the cosine dissimilarity function and m is a positive margin.

Unlike [17],[18] integrates word label information as input in training their NAWE model. The word labels are converted to character sequences $\mathbf{c} = [c_t; t \in [L]]$ of variable length L where c_i is a one-hot encoding of the i^{th} character in the word. A separate character based word embedding network g is used to generate a fixed dimensional vector embedding for input character sequence. A BiLSTM is used as network g which takes in character sequence C and outputs the last time step representation as word embedding $w = g(\mathbf{c}) = g(c_L)$.

[18] trains their model f with pairs (a, n) of dissimilar MFCC feature sequences and optimizes a multi-view loss function,

$$\mathcal{L}_{multi} = \mathcal{L}_1 + \mathcal{L}_2 \tag{3}$$

The first term in $\mathcal{L}_{multi}$ trains the anchor AWE $e_a = f(a)$ to discriminate anchor word $\mathbf{c}_a$ embedding $w_a = g(\mathbf{c}_a)$ from negative word $\mathbf{c}_n$ embedding $w_n = g(\mathbf{c}_n)$. It is defined as,

$$\mathcal{L}_1 = \max\{0, \widetilde{m} + d_{cos}(f(a), g(\mathbf{c}_a)) - d_{cos}(f(a), g(\mathbf{c}_n))\} \tag{4}$$

where $\widetilde{m}$ is a margin parameter. The second term in $\mathcal{L}_{multi}$ trains the anchor word embedding w_a to discriminate anchor AWE e_a from negative AWE $e_n = f(n)$. It is defined as,

$$\mathcal{L}_2 = \max\{0, m + d_{cos}(f(a), g(\mathbf{c}_a)) - d_{cos}(f(n), g(\mathbf{c}_a))\} \tag{5}$$

where m is a positive constant margin.

[19, 20] are SOTA approaches which train their model f with pairs (a, p). Such approaches don't necessarily need the word boundaries and word label information to generate pairs. Even in the case where the dataset only contains unlabelled audio, we can sample random audio segments and generate a positive for each segment via application of different augmentations and noise to it. Hence, such approaches can be trained totally un-supervised as well.

[19] trains their model f in encoder-decoder fashion. The NAWE encoder network f reads the input sequence while sequentially updating it's hidden states. A decoder network g generates a sequence of representations with the last time step encoder representation or NAWE $f(a)$. The model is first pre-trained to reconstruct the input MFCC a at each time step t with t-time step decoder representation $g_t(f(a))$ generated with NAWE $f(a)$. It is then trained on pairs (a, p) to reconstruct MFCC feature at t-time step of p with t-time step decoder representation $g_t(f(a))$. For both the tasks, mean squared error is used as the reconstruction loss.

[20] uses the contrastive task introduced in [26] where the targets are generated via simulation of MIPS over batches of pairs of AWEs of word segments corresponding to the same word, $B = \{e_{a_i}, e_{p_i}\}_{i=1}^N$ where N is the batch size. A mini-batch of NAWEs of size $2N$ is generated from batch B over which MIPS is conducted. A query NAWE from the mini-batch is sampled and via MIPS the NAWE in the mini-batch which gives the maximum cosine similarity score with query is returned. Given the information of which pairs of NAWEs actually correspond to similar word utterances from batch B, a contrastive task is defined with the self-supervision that for each NAWE e_{a_i}, the most similar NAWE in the mini-batch is e_{p_i}. The loss is defined as,

$$\mathcal{L}_{MIPS} = -\sum_{i=1}^{N} \log \frac{\exp(cos(e_{a_i}, e_{p_i}))}{\sum_{k \in \{a,p\}} \sum_{j=1}^{N} \exp(cos(e_{a_i}, e_{k_{j \neq i}}))} \tag{6}$$

4. Our Learning Approach

We train our NAWE model on batches of pairs of same word utterances (a, p) like [19, 20]. The word utterances in each pair in the batch are converted to MFCC sequences (X_a, X_p) of length T_a and T_p time steps respectively. We use the same NAWE implementation as [17, 18, 19, 20]. A BiLSTM encoder model takes in each MFCC sequence $X_{i \in \{a,p\}}$ and generates T_i length sequence of representations Z_i. The T_i-time step representation $z^i_{T_i} \in Z_i$ is used as AWE e_i corresponding to X_i giving the following equation for our NAWE model f,

$$f(X_i) = e_i = z^i_{T_i} \tag{7}$$

Our NAWE model is trained on pairs (X_a, X_p) in an encoder-decoder fashion like [19] but instead of reconstructing t time-step feature vector in each input sequence $X_{i\in\{a,p\}}$, we predict a noisy target or pseudo-label with t time-step decoder representation generated with NAWE $f(X_{j\neq i, j\in\{a,p\}}) = e_j$ of the other sequence X_j in the pair. The pseudo-label at each time-step in input sequence X_i is generated by a K-means clustering model h which is trained offline over MFCC features. We demonstrate our training approach in Figure 1.

A decoder BiLSTM g inputs AWE e_i generated from X_i and outputs a sequence of decoder representations time-aligned with the other sequence X_j in the pair. The sequence of decoder representations is given as $O_{ij} = [o_t^i = g_t(e_i); t \in [T_j]]$ where T_j is length of sequence X_j. At each time step $t \in [T_j]$, we use decoder representation $g_t(e_i)$ to predict the pseudo-label at that time-step in X_j .

4.1. Pseudo-label Generation

We train a K-Means clustering model h in an offline clustering step to generate pseudo-labels for each time step t of each sequence X_i in pair (X_a, X_p). Clustering model h generates target sequence $h(X_i) = \mathcal{T}_i = [\tau_t^i; t \in [T_i]]$ time aligned with X_i. Each target τ_t^i can be any label in finite collection $A = [K]$ of $K = 100$ pseudo-labels. We choose $K = 100$ as we want the pseudo-labels to be sub-phonetic. We did experiments with $K \in \{50, 100, 1000\}$ pseudo-labels. Increasing K brought only a slight improvement in performance.

4.2. Pairwise Prediction Task

For each time step $t \in [T_j]$, we use decoder representation $o_t^i \in O_{ij}$ to identify the true target $\tau_t^j \in \mathcal{T}_j$ at time step t for $x_t^j \in X_j$(see Fig. 1). The loss is defined as ,

$$\mathcal{L} = -\sum_{i,j\in\{a,p\}} \sum_{t=1}^{T_j} \log \frac{\exp cos(W o_t^i, v_{\tau_t^{j\neq i}})}{\sum_{\tau=1}^{K} \exp cos(W o_t^i, v_\tau)} \quad (8)$$

where $cos(,)$ is cosine distance, $W \in R^{d\times d'}$ is a projection matrix and $v_\tau \in R^{d'}$ is embedding corresponding to target τ.

v_τ is used to calculate the probability of decoder representation $o_t^i \in O_{ij}$ mapping to pseudo-label $\tau \in A$. Hence, we randomly initialize K such embeddings forming codebook $V = [v_k; k \in [K]]$ where member $v_k \in R^{d'}$ is an embedding representative of pseudo-label $k \in A$. Minimization of loss $\mathcal{L}$ tunes the encoder-decoder parameters as well as codebook V.

4.3. Iterative Refinement of Targets

We perform multiple iterations of training, with each iteration labelling the input word-segment MFCC sequences to our NAWE with a new generation of pseudo-labels. This performs iterative refinement of the pseudo-labels with each generation performing better semantic labelling of input MFCC sequences.

In the first iteration, we generate targets via K-means clustering over MFCC features. The clustering model h takes in MFCC X_i to generate target sequence $\mathcal{T}_i = h(X_i)$.

In further iterations of training, we generate new generation of targets via K-means clustering over the discriminatory decoder representations generated by encoder-decoder model trained in previous iteration. The K-means clustering model h in m^{th} training iteration now takes in decoder representation sequence $O_i^{m-1} = \{o_t^i; t \in [T_i]\}$ (generated by decoder trained in $m-1$ training iteration) time-aligned with input MFCC sequence X_i to generate target sequence $\mathcal{T}_i = h(O_i^{m-1})$.

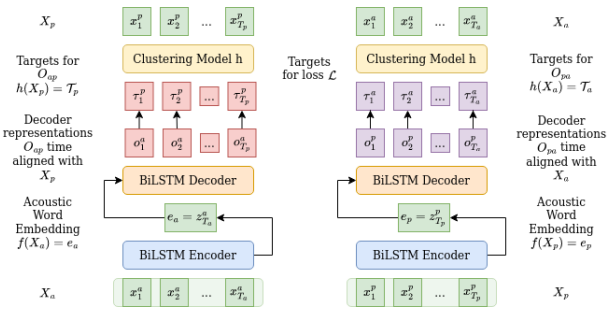

Figure 1: *Loss calculation. X_a and X_p are same word utterances but differ in rate, pitch, and voice. The targets for loss $\mathcal{L}$ are generated by clustering model h trained offline via K-Means clustering over MFCCs or pre-trained decoder representations.*

5. Experiments

We evaluate our approach to training NAWEs with approaches [17, 18, 19, 20] in generating discriminatory AWEs. We perform Word Discrimination task [22], which is used to for rapid evaluation of representations for STD, on TIMIT [27] and LibriSpeech [28] datasets. This task has been used in [18, 19, 20] to evaluate AWEs. We evaluate models trained with **weak supervision** i.e. word boundary and word label information is used to generate pairs or triplets for training, and **no supervision** i.e. trained on random $1s$ segments of audio.

Pairs of variable length word utterances are fed to the NAWE models which generate an AWE for each. A cosine distance is then calculated between the two AWEs to determine whether the pair of utterances are same word or not. We follow [18, 19, 20] in using AP calculated with ground-truth for evaluation. We conduct 10 trials of training and evaluation and have provided the mean AP and variance in our results (see Table 2).

5.1. Data

5.1.1. Training

Weak Supervision. We sample utterances of 887 words in the TIMIT [27] training split for training.The total number of distinct utterances used for training amount to $10.4K$ utterances. The number of characters in each of the words range from 3 to 15. Each word utterance range in duration from $0.25s$ to $1.17s$.
Un-supervised. We sample random $1s$ segments of audio from the LibriSpeech [28] 100 hours dataset. Similar pairs are generated via application of time-stretch and pitch-shift to audio.

5.1.2. Testing

We sample utterances of 887 words in the TIMIT [27] testing split disjoint from words used in training for evaluation of performances of the models on OOV words. We perform Word Discrimination task on $88.8K$ pairs of word utterances sampled from our test dataset. The evaluation dataset contains equal number of positive and negative pairs.

We further evaluate the generalizibility of the AWEs to different dataset. We conduct Word Discrimination task over $93.2K$ pairs of word utterances sampled from LibriSpeech [28] 100 hours dataset. A total of 1055 OOV words were used. The number of characters in each word vary in the range 3 to 12. The word boundaries were extracted via force alignment with transcription. The duration of each word utterance vary in the range $0.3s$ to $1.49s$.

5.2. Architecture Details and Hyper-parameters

We present the model architectures used in our experiments in Table 1. We use ADAM [28] optimizer and use a linear schedule of learning rate 0.001 with 8% of the training steps as warm-up.

Table 1: *BiLSTM Model Architectures*

Models	Layers	Hidden Cells	AWE dimension
Siamese Triplet [17]	3	256	512
Multi-View [18]	2	512	1024
EncDec-CAE [19]	3	256	512
ContrastiveRNN [20]	3	256	512
Ours	3	256	768

The margins used in eqs 2, 5 are set to 0.4. The margin parameter $\widetilde{m}$ used in eq 4 is implemented as,

$$\widetilde{m} = m_{max} \frac{\max\{d_{max}, editDistance(\mathbf{c}_a, \mathbf{c}_n)\}}{d_{max}} \quad (9)$$

where m_{max} is maximum margin set to 0.7 and d_{max} is the maximum distance between positive word and negative word in the dissimilar pairs used for training in [18].

5.3. Results

Table 2: *Word Discrimination (Average Precision)*

Models	TIMIT OOV	LibriSpeech OOV
MFCC + DTW	93	85
Weak Supervision (Trained on TIMIT)		
Siamese Triplet [17]	93.0 $\pm$0.8	78.3 $\pm$0.7
Multi-View [18]	92.5 $\pm$0.3	77.0 $\pm$1.4
EncDec-CAE [19]	96.2 $\pm$0.3	83.3 $\pm$0.4
ContrastiveRNN[20]	94.3 $\pm$0.4	82.4 $\pm$2.0
Ours	98.8 $\pm$0.03	86.4$\pm$0.1
No Supervision (Trained on LibriSpeech)		
EncDec-CAE [19]	80.8 $\pm$0.3	77.7 $\pm$0.4
ContrastiveRNN[20]	80.1 $\pm$0.4	75.1 $\pm$1.0
Ours	86.8 $\pm$0.13	81.4$\pm$0.2

Weak Supervision. We observe our approach to outperform all the baselines in generating discriminatory AWEs in terms of AP (see Tab.2) and Detection Error Trade-off (DET) curves (see Fig.2). All the NAWE models were trained on TIMIT and tested on TIMIT OOV and LibriSpeech OOV words. Our model presents SOTA performances on both datasets and generalises the most. Our approach generates better AWEs than [19] (SOTA, 2019). We present performances of raw MFCC where similarity scores between pairs are generated via DTW. We observe NAWEs to be more discriminative than MFCC (Fig. 2). We perform only 2 iterations of training and observe 1% performance increase with refinement of targets.

Un-supervised. Our approach performs the best in the unsupervised setting. Our model generates AWEs comparable to MFCC features in terms of Average Precision (see Tab.2) and DET curves (see Fig.2), while being trained on random $1s$ segments of audio. The models are evaluated on strictly OOV words sampled from TIMIT and LibriSpeech. Our NAWE model generalises well to different datasets.

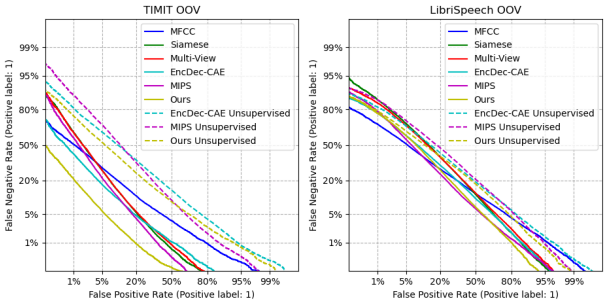

Figure 2: *DET Curves on TIMIT OOV and Librispeech OOV.*

Table 3: *Spoken Term Detection on TIMIT*

Models	ATWV	Search Time (s)
MFCC + S-DTW	0.55	0.002
EncDec-CAE [19] + S-RAILS	0.51	0.00009
Ours + S-RAILS	0.53	0.00009

Spoken Term Detection. We conduct STD on 1638 utterances in TIMIT test. We choose 612 words which occur atmost once in each of the utterances as queries to our detector system. We generate a query set $\mathcal{Q}$ with utterances of the query words extracted from our validation split of utterances in TIMIT train. For a given query $q \in \mathcal{Q}$ and utterance X, we detect and localize q in X. We implement a baseline STD system where we use S-DTW over MFCC features to detect and localize query. We split the search utterance to overlapping segments of a fixed length of $0.5s$. There is an overlap of $0.25s$ between two consecutive segments. The NAWE based STD systems converts query to AWE e_q and the utterance, now splitted to a sequence of overlapping segments to a sequence of AWEs $f(X) = [e_i; i \in [N]]$. We detect and localize query using S-RAILS with $b = 1024$ bits, $P = 16$ permutations and choose the top match as hit. We use word boundary information in each utterance to calculate Actual Term Weighted Value (ATWV) [29] (the larger the better). AWEs bring 2 orders of search time drop while giving comparable ATWV to MFCCs (see Tab.3). The search performance may be improved by splitting the utterance to overlapping segments from some minimum duration to some maximum duration (instead of fixed duration) and using S-RAILS to query over an enhanced dataset of AWEs.

6. Conclusion

In this paper, we present a novel pairwise learning approach to learning discriminative AWEs. Experiments on Word Discrimination and STD task show SOTA performance of our approach.

Decoder g representations and codebook of embeddings V may be used to perform labelling of acoustic segments by our model. Hence, we can convert the query and long utterance in our STD task to strings. We wish to explore if we can use fast text retrieval algorithms to perform detection of query.

The codes are available in `https://github.com/madhavlab/2023_adhiraj_encdecPairwisePred`.

7. References

[1] W. Shen, C. M. White, and T. J. Hazen, "A comparison of query-by-example methods for spoken term detection," MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, Tech. Rep., 2009.

[2] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for oov terms," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 404–409.

[3] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2007.

[4] C.-a. Chan and L.-s. Lee, "Model-based unsupervised spoken term detection with spoken queries," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1330–1342, 2013.

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

[6] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "High-performance query-by-example spoken term detection on the sws 2013 evaluation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7819–7823.

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

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[9] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 264–277, 2015.

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

[11] M. Müller, *Information retrieval for music and motion*. Springer, 2007, vol. 2.

[12] S. Bengio and G. Heigold, "Word embeddings for speech recognition," 2014.

[13] K. Levin, A. Jansen, and B. Van Durme, "Segmental acoustic indexing for zero resource keyword search," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5828–5832.

[14] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4950–4954.

[15] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Unsupervised learning of audio segment representations using sequence-to-sequence recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 765–769.

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

[17] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," *arXiv preprint arXiv:1706.03818*, 2017.

[18] W. He, W. Wang, and K. Livescu, "Multi-view recurrent neural acoustic word embeddings," *arXiv preprint arXiv:1611.04496*, 2016.

[19] H. Kamper, "Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6535–3539.

[20] C. Jacobs, Y. Matusevych, and H. Kamper, "Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 919–926.

[21] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[22] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[23] K. Levin, A. Jansen, and B. Van Durme, "Segmental acoustic indexing for zero resource keyword search," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5828–5832.

[24] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.

[25] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002, pp. 380–388.

[26] S. Chang, D. Lee, J. Park, H. Lim, K. Lee, K. Ko, and Y. Han, "Neural audio fingerprint for high-specific audio retrieval based on contrastive learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3025–3029.

[27] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] J. Fiscus, J. Ajot, and G. Doddington, "The spoken term detection (std) 2006 evaluation plan," *NIST USA, Sep*, vol. 86, 2006.